Matematica delle Reti Neurali

Volume X

Fisica Statistica Profonda · *Topologia Applicata* · *Active Learning Bayesiano*

Teoria delle Matrici Casuali · Metodo delle Repliche · Spin Glass e Transformer · Omologia Persistente · Numeri di Betti del Loss Landscape · Algoritmo Mapper · Campionamento Informativo · Query-by-Committee · Optimal Experimental Design

---

Silvano Salvador

Continuazione dei Volumi I–IX (Sezioni 0–542)

Sezioni 543–640 · Derivazioni complete · Ogni simbolo definito · Esempi numerici espliciti

Nota metodologica: La notazione segue la Sezione 0 del Volume I. Ogni simbolo è ridefinito localmente alla prima occorrenza. Le dimostrazioni complete sono marcate con □ e possono essere saltate in prima lettura senza perdere il filo della trattazione. I rimandi ai volumi precedenti sono espliciti e corredati di una breve ricapitolazione del concetto rilevante.

Abstract

Il presente volume costituisce la continuazione autonoma dei Volumi I–IX della serie *Matematica delle Reti Neurali*. Copre sessanta sezioni principali (Sezioni 543–602) organizzate in tre parti tematiche, trattando le frontiere matematiche più profonde del deep learning con rigore uniforme: ogni affermazione è fondata su una derivazione esplicita, ogni simbolo è definito nel Glossario (Appendice A).

I contributi principali del volume includono: il **framework della fisica statistica applicata al deep learning**, con la teoria spettrale delle matrici di peso via random matrix theory (RMT), il metodo delle repliche di Parisi applicato alle reti neurali, e la formalizzazione della spin glass theory per l'analisi dei minimi del loss landscape nei Transformer; la **topologia algebrica applicata al deep learning**, con la costruzione rigorosa dell'omologia persistente, il calcolo dei numeri di Betti del loss landscape e degli spazi di rappresentazione, e l'algoritmo Mapper come strumento di visualizzazione topologica; infine l'**active learning e il Bayesian experimental design**, con la matematica dell'uncertainty sampling, del query-by-committee, dell'expected information gain, e dell'optimal experimental design bayesiana, in connessione con i Gaussian Process trattati nel Volume V.

Indice

Abstract **2**

Prefazione **8**

1 Sezione A — Fisica Statistica Profonda e Reti Neurali **9**
1.1 §543 Spettro delle matrici di peso: Random Matrix Theory . . . 9
1.1.1 Legge semicircolare di Wigner . . . 9
1.1.2 Legge di Marchenko-Pastur per matrici rettangolari . . . 9
1.2 §544 Diagnostica dello spettro in reti addestrate . . . 10
1.2.1 Il fenomeno Heavy-Tail Self-Regularization . . . 10
1.3 §545 Il metodo delle repliche . . . 10
1.3.1 Setup del modello di Gardner . . . 10
1.3.2 Il trucco delle repliche . . . 11
1.4 §546 Rottura della simmetria delle repliche e ultrametricity . . . 11
1.5 §547 Spin Glass e Transformer . . . 12
1.5.1 Il modello SK applicato all'attenzione . . . 12
1.5.2 Free energy e convergenza dell'attenzione . . . 12
1.6 §548 Analisi spettrale del NTK via RMT . . . 13
1.7 §549–§552 Applicazioni: pruning spettrale e diagnostica della compressione . . . 13
1.7.1 §549 Singular Value Decomposition e rango effettivo . . . 13
1.7.2 §550 Pruning spettrale low-rank . . . 14
1.7.3 §551 Stabilità spettrale durante il fine-tuning . . . 14
1.7.4 §552 Riepilogo della Sezione A1 . . . 14

2 Sezione B — Topologia Applicata al Deep Learning **15**
2.1 §568 Fondamenti di topologia algebrica . . . 15
2.1.1 Complessi simpliciali . . . 15
2.1.2 Omologia simpliciale . . . 15
2.2 §569 Omologia persistente . . . 16
2.2.1 Filtrazione di Vietoris-Rips . . . 16
2.2.2 Diagrammi di persistenza . . . 16
2.3 §570 Numeri di Betti del loss landscape . . . 16
2.3.1 Costruzione della sublevel filtration . . . 16
2.3.2 Interpretazione per il training di reti neurali . . . 17
2.4 §571 L'algoritmo Mapper . . . 17
2.4.1 Mapper sulle rappresentazioni neurali . . . 18
2.5 §572 Distanza di Wasserstein tra diagrammi di persistenza . . . 18
2.6 §573–§577 Betti del loss landscape nei Transformer . . . 18
2.6.1 §573 Topologia delle rappresentazioni di layer . . . 18
2.6.2 §574 Persistent homology del training . . . 19
2.6.3 §575 Topologia e generalizzazione . . . 19
2.6.4 §576 Calcolo efficiente dell'omologia persistente . . . 19
2.6.5 §577 Riepilogo della Sezione B . . . 20

3 Sezione C — Active Learning e Bayesian Experimental Design **21**
3.1 §588 Formulazione del problema di active learning . . . 21
3.2 §589 Uncertainty Sampling . . . 21
3.3 §590 Query-by-Committee (QBC) . . . 22
3.3.1 QBC bayesiano via approssimazione della posterior . . . 22
3.4 §591 Expected Information Gain (EIG) . . . 23

3.4.1 BALD: Bayesian Active Learning by Disagreement . . . 23
3.5 §592 Optimal Experimental Design bayesiano . . . 23
3.5.1 OED bayesiano non lineare . . . 24
3.6 §593 Connessione con i Gaussian Process . . . 24
3.7 §594 Active learning su sequenze e LLM . . . 25
3.7.1 EIG per instruction tuning . . . 25
3.8 §595 Batch active learning e submodularità . . . 25
3.9 §596–§598 Core-Set e diversità . . . 26
3.9.1 §596 Il metodo Core-Set . . . 26
3.9.2 §597 PowerBALD: batch EIG con diversità . . . 26
3.9.3 §598 Confronto sistematico delle strategie . . . 27
3.10 §599 Connessione con il Bayesian Experimental Design per reti neurali profonde 27
3.11 §600 Teorema di no-free-lunch per l'active learning . . . 27
3.12 §601 Complessità del campionamento e legami con la VC theory . . . 28
3.13 §602 Riepilogo della Sezione C e del ciclo . . . 28

A Appendice A — Glossario dei Simboli Principali 29

B Appendice B — Raccolta dei Teoremi Principali 30

C Appendice C — Note bibliografiche 31

Nota conclusiva 32

D Sezione A (continuazione) — Free Probability, Equazioni TAP e RFOT 32
D.1 §553 Free Probability Theory . . . 32
D.1.1 Libera indipendenza . . . 32
D.1.2 Trasformata R e addizione libera . . . 32
D.1.3 Trasformata S e moltiplicazione libera . . . 33
D.2 §554 Equazioni di TAP e campo medio per reti neurali . . . 33
D.2.1 Campo medio naive e termine di Onsager . . . 33
D.2.2 Equazioni TAP come message passing . . . 33
D.3 §555 RFOT: Random First-Order Transition Theory . . . 34
D.3.1 Transizioni di fase del loss landscape . . . 34
D.4 §556 Analisi spettrale della Hessiana . . . 35
D.4.1 Decomposizione della Hessiana: GGN + residuo . . . 35
D.4.2 Distribuzione spettrale della Hessiana via RMT . . . 35
D.5 §557–§567 Teoria mean-field per reti profonde . . . 36
D.5.1 §557 Propagazione del segnale: teoria mean-field dei layer . . . 36
D.5.2 §558 Correlazione tra input: depth scale . . . 36
D.5.3 §559 Connessione con il NTK e regimi di training . . . 36
D.5.4 §560 Derivazione mean-field per Transformer . . . 37
D.5.5 §561 Benign overfitting e interpolation threshold . . . 37
D.5.6 §562–§567 Sintesi: dalla fisica statistica al design architetturale . . . 38

E Sezione B (continuazione) — TDA Avanzata: Čech, Zigzag e Sheaves 39
E.1 §578 Il complesso di Čech e il teorema dei nervi . . . 39
E.1.1 Confronto Čech vs Vietoris-Rips . . . 39
E.2 §579 Omologia persistente zigzag . . . 39
E.2.1 Applicazione: analisi temporale delle rappresentazioni . . . 40
E.3 §580 Teoria di Morse e funzioni di filtraggio . . . 40
E.3.1 Funzioni di filtro per il deep learning . . . 41
E.4 §581 Omologia persistente multidimensionale . . . 41

E.5 §582 Sheaf Theory e reti neurali . . . 41
E.6 §583–§587 TDA per LLM: applicazioni concrete . . . 42
E.6.1 §583 Topologia degli attention head . . . 42
E.6.2 §584 Betti degli spazi di rappresentazione per classi diverse . . . 42
E.6.3 §585 TDA per la detecting delle allucinazioni . . . 42
E.6.4 §586 TDA e compressione: pruning topologicamente guidato . . . 43
E.6.5 §587 Riepilogo Sezione B avanzata . . . 43

F Sezione C (continuazione) — Ottimizzazione Bayesiana, Neural Bandits e Fondamenti PAC 44
F.1 §603 Ottimizzazione Bayesiana (BO) . . . 44
F.1.1 Acquisition functions . . . 44
F.1.2 Regret del kernel e complessità dell'ottimizzazione . . . 45
F.2 §604 Neural Bandits e Thompson Sampling . . . 45
F.2.1 Thompson Sampling neurale . . . 45
F.3 §605 Agnostic Active Learning . . . 46
F.4 §606 Active Learning per structured prediction . . . 46
F.5 §607 Core-Set avanzato: k-medoids e Facility Location . . . 47
F.5.1 k-medoids come CoreSet robusta . . . 47
F.6 §608 Self-supervised active learning . . . 48
F.7 §609 Multi-fidelity active learning . . . 48
F.8 §610 Connessione: active learning, informazione di Fisher e geometria dell'informazione . . . 49
F.9 §611 Regret bayesiano e ottimalità minimax . . . 49
F.10 §612 Riepilogo finale e visione unificante . . . 50

G Appendice D — Connessioni tra i Volumi 51

H Appendice E — Dimostrazione: Additività della trasformata R 51

I Appendice F — Algoritmo Ripser per il calcolo efficiente dell'omologia persistente 51

J Appendice G — Codice: Omologia persistente su rappresentazioni di BERT 52

K Appendice H — Tavola delle acquisition functions 52

Nota conclusiva 53

L Sezione D — PAC-Bayes per l'Active Learning e Generalizzazione Bayesiana 53
L.1 §613 Il framework PAC-Bayes . . . 53
L.1.1 PAC-Bayes e reti neurali . . . 53
L.2 §614 PAC-Bayes per l'active learning . . . 54
L.3 §615 Connessione PAC-Bayes e Information Bottleneck . . . 54
L.4 §616 Rademacher Complexity e Active Learning . . . 55

M Sezione E — Topological Signal Processing e Reti Neurali su Complessi 56
M.1 §617 Topological Signal Processing (TSP) . . . 56
M.2 §618 Simplicial Neural Networks . . . 57
M.3 §619 Convoluzione di Hodge e filtri topologici . . . 57
M.4 §620 Topological Persistence e Neural Collapse . . . 58

N Sezione F — Sintesi: Visione Geometrica dell'Apprendimento **59**
N.1 §621 Il triangolo fondamentale: Geometria, Topologia, Statistica 59
N.2 §622 Ricapitolazione finale: 60 sezioni in 10 teoremi 60

O Sezione G — Lottery Ticket Hypothesis e Teoria della Compressione **61**
O.1 §623 La Lottery Ticket Hypothesis 61
O.1.1 Algoritmo Iterative Magnitude Pruning (IMP) 61
O.1.2 Basi teoriche: connessione con la teoria dei circuiti 61
O.2 §624 Connessione LTH con la Teoria di Morse 62
O.3 §625 Distillazione della conoscenza e compressione teorica 62
O.4 §626 Complessità di Kolmogorov e MDL 63

P Sezione H — Optimal Transport e Active Learning **64**
P.1 §627 Fondamenti dell'Optimal Transport 64
P.2 §628 Distanza di Wasserstein come metrica di active learning 64
P.2.1 Sinkhorn e active learning scalabile 64
P.3 §629 Frechet Inception Distance e qualità generativa come active learning . . 65
P.4 §630 Sliced Wasserstein e scalabilità 65

Q Sezione I — TDA, Geometria dell'Informazione e Connessioni Profonde **67**
Q.1 §631 Varietà statistica e omologia persistente 67
Q.1.1 Connessione lineare tra minimi: Loss Valley 67
Q.2 §632 Divergenze di Rényi e generalizzazioni della persistenza 67
Q.3 §633 Optimal Transport e Apprendimento per Trasferimento 68

R Sezione J — Reti a Larghezza Infinita e Connessione con i Gaussian Process **69**
R.1 §634 Il limite gaussiano delle reti neurali 69
R.1.1 Kernel per attivazioni comuni 69
R.2 §635 NNGP vs NTK: due regimi di larghezza infinita 69
R.3 §636 Active Learning ottimale nel regime NNGP 70
R.4 §637 Connessioni PAC-Bayes nel regime NTK 71

S Sezione K — Sintesi Numerica: Tre Casi di Studio Completi **72**
S.1 §638 Caso I: Analisi spettrale completa di GPT-2 small 72
S.1.1 Diagnostica RMT layer per layer 72
S.1.2 Connessione con la perplexity 72
S.2 §639 Caso II: Topologia del training di BERT su SST-2 72
S.2.1 Protocollo sperimentale 72
S.2.2 Risultati 73
S.3 §640 Caso III: Active Learning end-to-end su NLP con budget limitato 73
S.3.1 Pipeline completa 73
S.3.2 Confronto con baseline 73

T Appendice I — Approfondimento: Equazioni TAP e Compressed Sensing **74**
T.1 Dualità tra AMP e l-1 minimization 74
T.2 State evolution e fasi del recupero sparso 74

U Appendice J — Dimostrazione: Regret di GP-UCB **75**

V Appendice K — Strumenti software e librerie **75**

W Appendice L — Tavola cronologica dei risultati **76**

Bibliografia **77**

Prefazione al Volume X

I nove volumi precedenti di questa serie hanno costruito un panorama matematico eccezionalmente ampio: dalle fondamenta del neurone artificiale (Vol. I) fino ai Grandi Modelli Linguistici e al loro allineamento (Vol. VII), passando per la geometria differenziale dell'informazione, la causalità, le leggi di scala, le strutture equivarianti, e infine la biologia computazionale con AlphaFold 2, AlphaFold 3, AlphaGenome ed ESM-3 (Vol. VIII). Il Volume IX ha trattato l'AI agentiva, il *inference-time compute scaling* e la generazione video.

Il presente volume conclude il ciclo colmando tre lacune specificamente indicate nella valutazione critica della serie:

1. **Fisica statistica e reti neurali: connessioni più profonde.** Il ciclo aveva toccato il Neural Tangent Kernel, il mean field e le Modern Hopfield Networks, ma la teoria delle matrici casuali per lo studio degli spettri dei pesi, il metodo delle repliche e la spin glass theory applicata ai Transformer restavano trattati marginalmente rispetto alla profondità matematica disponibile.

2. **Topologia applicata al deep learning.** La Topological Data Analysis (TDA), l'omologia persistente applicata alle rappresentazioni neurali (numeri di Betti del loss landscape, algoritmo Mapper) è un'area con risultati formali importanti completamente assente dal ciclo.

3. **Active learning e Bayesian experimental design.** La matematica del campionamento informativo (uncertainty sampling, query-by-committee, expected information gain, optimal experimental design bayesiana) era quasi assente, nonostante le sue connessioni profonde con i Gaussian Process trattati nel Volume V.

Tesi centrale del Volume X. *L'apprendimento automatico rigoroso richiede non solo l'algebra lineare e il calcolo differenziale dei volumi precedenti, ma anche strumenti della fisica statistica (per capire la geometria del loss landscape a grande scala), della topologia algebrica (per caratterizzarne la struttura globale invariante per deformazione) e della statistica bayesiana decisionale (per acquisire dati in modo ottimale). Questi tre filoni, apparentemente distinti, convergono in una visione unificata dell'apprendimento come inferenza geometrica e statistica.*

Struttura del volume.

- **Sezione A** (§543–§567): Fisica statistica profonda — teoria spettrale RMT, metodo delle repliche, spin glass e Transformer.
- **Sezione B** (§568–§587): Topologia applicata — complessi simpliciali, omologia persistente, diagrammi di persistenza, numeri di Betti del loss landscape, algoritmo Mapper.
- **Sezione C** (§588–§602): Active learning e Bayesian experimental design — uncertainty sampling, QBC, EIG, OED, connessioni con i GP.
- **Appendici**: Glossario, tavola dei simboli, raccolta dei teoremi principali, note bibliografiche.

Riferimento al Volume I

I simboli e le notazioni seguono fedelmente la Sezione 0 del Volume I (Parte 0 — Notazioni e Prerequisiti). In caso di dubbio su un simbolo, si rimanda a quella sezione. I simboli

aggiuntivi specifici di questo volume sono definiti nell'Appendice A.

1 Sezione A — Fisica Statistica Profonda e Reti Neurali

1.1 §543 Spettro delle matrici di peso: Random Matrix Theory

I pesi di una rete neurale profonda, al termine del training, non sono matrici casuali generiche. Eppure la *Random Matrix Theory* (RMT) fornisce una lente sorprendentemente potente per capire cosa distingue una rete *ben addestrata* da una casuale, e perché certi spettri segnalano una capacità di generalizzazione elevata.

Definizione 1.1: Ensemble di Wigner

Una **matrice di Wigner** $\mathbf{W} \in \mathbb{R}^{n\times n}$ è una matrice simmetrica con voci i.i.d. (a parte la simmetria)

$$W_{ij} = W_{ji}, \quad \mathbb{E}[W_{ij}] = 0, \quad \mathbb{E}[W_{ij}^2] = \frac{\sigma^2}{n}, \quad i \le j.$$

L'ensemble di Wigner *GOE* (Gaussian Orthogonal Ensemble) corrisponde a $W_{ij} \sim \mathcal{N}(0, \sigma^2/n)$ per $i < j$ e $W_{ii} \sim \mathcal{N}(0, 2\sigma^2/n)$.

1.1.1 Legge semicircolare di Wigner

Teorema 1.1: Legge semicircolare di Wigner

Sia $\mathbf{W}_n$ una sequenza di matrici di Wigner con varianza σ^2. La **distribuzione spettrale empirica**

$$\mu_n \triangleq \frac{1}{n}\sum_{i=1}^{n} \delta_{\lambda_i(\mathbf{W}_n)}$$

converge quasi certamente, in senso debole, alla **legge semicircolare**

$$\rho_{sc}(\lambda) = \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2 - \lambda^2}\,\mathbb{1}_{|\lambda|\le 2\sigma}.$$

Dimostrazione. Il metodo dei momenti mostra che $\mathbb{E}[\mathrm{tr}(\mathbf{W}_n^{2k})/n] \to C_k \sigma^{2k}$ dove $C_k = \binom{2k}{k}/(k+1)$ sono i numeri di Catalan, che corrispondono esattamente ai momenti della misura semicircolare. La convergenza quasi certa segue da una stima di concentrazione via il lemma di Borel-Cantelli. □ □

1.1.2 Legge di Marchenko-Pastur per matrici rettangolari

Le reti neurali usano matrici di peso $\mathbf{W} \in \mathbb{R}^{m\times n}$. Sia $\mathbf{X} \in \mathbb{R}^{n\times p}$ una matrice di dati casuali e $\mathbf{S} = \mathbf{X}\mathbf{X}^\top/p$ la sua matrice di covarianza campionaria. Il rapporto $\gamma = n/p$ è detto *aspect ratio*.

Teorema 1.2: Legge di Marchenko-Pastur

Se $\mathbf{X}_{ij} \overset{iid}{\sim} \mathcal{N}(0,\sigma^2)$ e $n/p \to \gamma \in (0,\infty)$, la distribuzione spettrale empirica di $\mathbf{S}$ converge alla **legge di Marchenko-Pastur**

$$\rho_{MP}(\lambda) = \frac{1}{2\pi\sigma^2}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\gamma\lambda}\cdot \mathbb{1}_{[\lambda_-,\lambda_+]}(\lambda),$$

dove $\lambda_\pm = \sigma^2(1 \pm \sqrt{\gamma})^2$.

La legge di Marchenko-Pastur è il *rumore di fondo* spettrale per le reti neurali. Gli autovalori che **emergono** sopra λ_+ sono detti *spike* e corrispondono a direzioni informativamente rilevanti.

Definizione 1.2: Numero di condizionamento spettrale

Per una matrice di peso $\mathbf{W} \in \mathbb{R}^{m\times n}$, lo **stable rank**

$$\mathtt{sr}(\mathbf{W}) \triangleq \frac{\|\mathbf{W}\|_F^2}{\|\mathbf{W}\|_2^2} = \frac{\sum_i \sigma_i^2}{\sigma_{\max}^2}$$

quantifica il numero effettivo di direzioni in cui $\mathbf{W}$ opera non-trivialmente. Per una matrice casuale GOE con $m = n$, $\mathtt{sr} \approx n/3$.

1.2 §544 Diagnostica dello spettro in reti addestrate

1.2.1 Il fenomeno Heavy-Tail Self-Regularization

Martin e Mahoney (2021) hanno osservato che i pesi di reti ben addestrate (GPT-3, BERT, ResNet) mostrano code pesanti nella distribuzione degli autovalori della matrice di covarianza $\mathbf{W}^\top\mathbf{W}$: la distribuzione segue una *power law*

$$\rho(\lambda) \propto \lambda^{-\alpha}, \quad \alpha \in [2, 4],$$

invece della legge di Marchenko-Pastur con decadimento esponenziale atteso per pesi casuali.

Proposizione 1.1: Coda pesante come indicatore di generalizzazione

Sia $\hat{\alpha}$ l'esponente power-law stimato dalla distribuzione spettrale di $\mathbf{W}^\top\mathbf{W}$ per un layer di una rete. Allora:

1. Reti non addestrate: $\hat{\alpha} \to \infty$ (distribuzione di Marchenko-Pastur, senza coda pesante).
2. Reti addestrate ma non convergenti: $\hat{\alpha} \approx 6$–8.
3. Reti ben convergenti con buona generalizzazione: $\hat{\alpha} \approx 2$–4.

ESEMPIO NUMERICO — BERT-large

Per BERT-large (Devlin et al., 2019) i layer di attenzione mostrano $\hat{\alpha} \approx 2.1$–3.4. Se si re-inizializzano casualmente gli stessi layer, $\hat{\alpha}$ sale immediatamente sopra 8, e l'accuracy su GLUE decade di ~ 15 punti percentuali, confermando la correlazione tra coda pesante e capacità di generalizzazione.

1.3 §545 Il metodo delle repliche

Il metodo delle repliche è una tecnica della fisica statistica, sviluppata da Parisi (1979) nel contesto dei vetri di spin, che permette di calcolare la *free energy media* di sistemi disordinati. Nel contesto del machine learning, il "disordine" è il dataset casuale; la "free energy" è legata alla loss media.

1.3.1 Setup del modello di Gardner

Consideriamo il *perceptron sferico*: un modello con N parametri $\boldsymbol{w} \in S^{N-1}$ (sfera unitaria), P esempi $(\boldsymbol{x}^\mu, y^\mu)$ con $\boldsymbol{x}^\mu \in \mathbb{R}^N$, e vincoli di classificazione corretta:

$$y^\mu \frac{\boldsymbol{w}^\top \boldsymbol{x}^\mu}{\sqrt{N}} \geq \kappa, \quad \mu = 1, \dots, P.$$

Il rapporto $\alpha = P/N$ è detto *carico*.

Definizione 1.3: Volume di Gardner

Il **volume di Gardner** è la misura del set di soluzioni:

$$V(\kappa, \alpha) \triangleq \int_{S^{N-1}} \prod_{\mu=1}^{P} \Theta\left(y^\mu \frac{\boldsymbol{w}^\top \boldsymbol{x}^\mu}{\sqrt{N}} - \kappa \right) \mathrm{d}\boldsymbol{w},$$

dove Θ è la funzione di Heaviside.

1.3.2 Il trucco delle repliche

Calcolare $\mathbb{E}_{\mathbf{X},\boldsymbol{y}}[\ln V]$ è difficile perché il logaritmo è dentro l'aspettazione. Il *trucco delle repliche* sfrutta l'identità:

$$\ln V = \lim_{n \to 0} \frac{V^n - 1}{n}.$$

Per n intero, $V^n = \int \prod_{a=1}^{n} \mathrm{d}\boldsymbol{w}^{(a)} \prod_\mu \prod_a \Theta(\cdots)$ è una funzione di n "copie" (*repliche*) dei parametri. Il calcolo per n intero, seguito dal limite $n \to 0$, produce la *replica-symmetric free energy*.

Teorema 1.3: Free energy replica-symmetric (Gardner, 1988)

Nell'ansatz di simmetria delle repliche (RS), la free energy media del perceptron sferico è

$$f_{RS}(\kappa, \alpha) = \operatorname{extr}_{q, \hat{q}} \left[\frac{q\hat{q}}{2} - \frac{\alpha}{2} \ln(1 - q) + \alpha \int \mathcal{D}u \, \ln H\left(\frac{\kappa - \sqrt{q}\, u}{\sqrt{1 - q}} \right) \right],$$

dove $q \in [0, 1]$ è il **parametro d'ordine** (overlap tra repliche), $H(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} \, \mathrm{d}t$ è il Q-function, e $\mathcal{D}u = (2\pi)^{-1/2} e^{-u^2/2} \, \mathrm{d}u$.

ESEMPIO NUMERICO — Capacità critica di Gardner

Per $\kappa = 0$ (classificazione senza margine), la soluzione replica-symmetric predice la **capacità critica di Gardner** $\alpha_c = 2$: per $\alpha < 2$ esistono con probabilità 1 soluzioni; per $\alpha > 2$ il sistema è in media sovradeterminato. Questo corrisponde alla bound $P < 2N$ per la memorizzazione nel perceptron, confermata empiricamente e dalla VC theory ($\alpha_c^{VC} = 2$).

1.4 §546 Rottura della simmetria delle repliche e ultrametricity

La soluzione RS di Parisi è esatta per sistemi convessi (perceptron sferico con $\kappa \geq 0$). Per sistemi con *paesaggi energetici* complessi (reti profonde), la simmetria delle repliche è rotta (RSB, Replica Symmetry Breaking).

Definizione 1.4: Parametro d'ordine di Parisi

La soluzione di Parisi introduce una **funzione** $q(x)$ per $x \in [0,1]$ (al posto dello scalare q RS), definita come

$$q(x) = \lim_{n\to 0} \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[w_i^{(a)} w_i^{(b)}\right]$$

per due repliche (a,b) con overlap inferiore alla frazione x degli stati. La funzione $q(x)$ è non-decrescente; il caso RS corrisponde a $q(x) = q$ costante.

La RSB implica una struttura *ultrametrica* dello spazio delle soluzioni:

Proposizione 1.2: Ultrametricity dei minimi

In un sistema con RSB di Parisi, i minimi locali della loss sono organizzati ultrametricamente: per ogni terna di soluzioni $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \boldsymbol{w}^{(3)}$,

$$d\Big(\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(3)}\Big) \leq \max\Big(d\Big(\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}\Big),\, d\Big(\boldsymbol{w}^{(2)}, \boldsymbol{w}^{(3)}\Big)\Big),$$

dove $d(\boldsymbol{w}, \boldsymbol{w}') = 1 - \frac{\boldsymbol{w}\cdot\boldsymbol{w}'}{N}$ è la distanza di Hamming normalizzata.

1.5 §547 Spin Glass e Transformer

I modelli di spin glass (Sherrington-Kirkpatrick, 1975) descrivono sistemi con interazioni disordinate. Il Transformer può essere visto come un sistema di spin glass nel seguente senso.

1.5.1 Il modello SK applicato all'attenzione

Consideriamo il meccanismo di attenzione scalata come un campo medio: la matrice di attenzione $\mathbf{A} \in \mathbb{R}^{N\times N}$ è

$$A_{ij} = \frac{e^{s_{ij}/\sqrt{d}}}{\sum_k e^{s_{ik}/\sqrt{d}}}, \quad s_{ij} = \boldsymbol{q}_i^\top \boldsymbol{k}_j.$$

Interpretiamo s_{ij} come un'interazione disordinata tra il token i e il token j: se i vettori $\boldsymbol{q}, \boldsymbol{k}$ sono casuali (come nella fase di pre-training), s_{ij} ha la distribuzione di un'accoppiamento SK.

Definizione 1.5: Modello SK per l'attenzione

La **Hamiltoniana di Sherrington-Kirkpatrick** per un sistema di N spin $\sigma_i \in \{-1, +1\}$ è

$$H_{SK}(\boldsymbol{\sigma}) = -\sum_{i<j} J_{ij}\sigma_i\sigma_j, \quad J_{ij} \overset{iid}{\sim} \mathcal{N}\left(0, \frac{J^2}{N}\right).$$

L'analogia con l'attenzione è: i token svolgono il ruolo di spin, e i punteggi $s_{ij}/\sqrt{d}$ svolgono il ruolo di J_{ij}.

1.5.2 Free energy e convergenza dell'attenzione

Teorema 1.4: Free energy SK (Parisi, 1979 — versione attentiva)

La free energy del modello SK di Parisi, nel limite termodinamico $N \to \infty$, è

$$f_{SK}(\beta) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}\left[-\ln Z_N(\beta)\right], \quad Z_N = \sum_{\boldsymbol{\sigma}} e^{-\beta H_{SK}(\boldsymbol{\sigma})}.$$

La soluzione esatta di Parisi (1979, 1983), dimostrata rigorosamente da Guerra-Toninelli (2002) e Talagrand (2006), è data da una funzione $q(x)$ soddisfacente l'equazione di Parisi:

$$\frac{\partial}{\partial x}\Phi(q(x),x) = \frac{J^2\beta^2}{2}\frac{\partial q}{\partial x}\left(\frac{\partial^2\Phi}{\partial q^2}\right)^2,$$

dove Φ è il *potenziale di Parisi*.

ESEMPIO NUMERICO — Transizione di fase nel self-attention

Per un layer di self-attention con N token, $d_k = 64$ e pesi di proiezione casuali $\sim \mathcal{N}(0, 1/d_k)$:

- A temperatura $\beta^{-1} = \sqrt{d_k}$ (il valore di default usato da Vaswani et al.), il sistema è nella fase *paramagnetica*: ogni token è attento a tutti in modo uniforme.
- Abbassando la temperatura (aumentando β), il sistema entra nella fase *spin glass*: l'attenzione si localizza su pochi token, corrispondente all'apprendimento di pattern.

La transizione avviene a $\beta_c = 1/J$ con $J^2 = d^{-1}$.

1.6 §548 Analisi spettrale del NTK via RMT

Il Neural Tangent Kernel (NTK), introdotto nel Volume II (§88), descrive la dinamica di training nel regime di larghezza infinita. La RMT fornisce la distribuzione degli autovalori del NTK.

Definizione 1.6: Kernel Gram del NTK

Per una rete con P esempi di training $\{\boldsymbol{x}^\mu\}_{\mu=1}^P$, la **matrice Gram del NTK** è

$$\mathbf{K}^{NTK} \in \mathbb{R}^{P\times P}, \quad K^{NTK}_{\mu\nu} = \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}^\mu),\, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}^\nu)\rangle\,.$$

La dinamica di gradient flow è $\dot{\boldsymbol{e}} = -\mathbf{K}^{NTK}\boldsymbol{e}$ dove $\boldsymbol{e}_\mu = f(\boldsymbol{x}^\mu) - y^\mu$ è il residuo.

Proposizione 1.3: Distribuzione spettrale del NTK per MLP wide

Per un MLP fully-connected con L layer, larghezza $n \to \infty$, e P esempi i.i.d. su S^{d-1}, la distribuzione spettrale empirica di $\mathbf{K}^{NTK}/n$ converge a una misura deterministica μ_{NTK} che soddisfa l'equazione di Marchenko-Pastur generalizzata:

$$s_{\mu_{NTK}}(z) = -\frac{1}{z + c\, s_{\mu_{NTK}}(z)\,\kappa_{NTK}},$$

dove $s_\mu(z) = \int (\lambda - z)^{-1}\,\mathrm{d}\mu(\lambda)$ è la trasformata di Stieltjes, $c = P/n$ e κ_{NTK} è un valore che dipende dall'architettura e dall'attivazione.

1.7 §549–§552 Applicazioni: pruning spettrale e diagnostica della compressione

1.7.1 §549 Singular Value Decomposition e rango effettivo

Sia $\mathbf{W} \in \mathbb{R}^{m\times n}$ una matrice di peso con SVD $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Il **rango effettivo** (Roy & Vetterli, 2007) è

$$\texttt{erank}(\mathbf{W}) \triangleq \exp\left(H\left(\frac{\sigma_1}{\|\boldsymbol{\sigma}\|_1}, \frac{\sigma_2}{\|\boldsymbol{\sigma}\|_1}, \dots\right)\right) = \exp\left(-\sum_i p_i \ln p_i\right),$$

dove $p_i = \sigma_i/\|\boldsymbol{\sigma}\|_1$ e H è l'entropia di Shannon. A differenza del rango algebrico, `erank` è continuo e misura quante direzioni singolari sono "sostanzialmente attive".

1.7.2 §550 Pruning spettrale low-rank

[Pruning low-rank adattivo]

Require: Matrice di peso $\mathbf{W}$, soglia $\epsilon > 0$

1: Calcola SVD: $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$

2: Ordina $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$

3: Trova il rango minimo $k^* = \min\left\{k : \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \geq 1-\epsilon\right\}$

4: **return** $\mathbf{W}_{k^*} = \mathbf{U}_{:,1:k^*}\,\mathbf{\Sigma}_{1:k^*,1:k^*}\,\mathbf{V}^\top_{:,1:k^*}$

Proposizione 1.4: Errore di approssimazione low-rank

L'errore di Frobenius dell'approssimazione rank-k^* è

$$\|\mathbf{W} - \mathbf{W}_{k^*}\|_F^2 = \sum_{i=k^*+1}^{r} \sigma_i^2 \leq \epsilon\|\mathbf{W}\|_F^2.$$

Questo è il teorema di Eckart-Young-Mirsky, ed è ottimale tra tutte le approssimazioni di rango $\leq k^*$.

1.7.3 §551 Stabilità spettrale durante il fine-tuning

Si può monitorare la "deriva" spettrale di ogni layer durante il fine-tuning con la *distanza di Wasserstein* tra le distribuzioni spettrali:

$$W_2(\mu_t, \mu_0) = \left(\int_0^1 \left(F_{\mu_t}^{-1}(p) - F_{\mu_0}^{-1}(p)\right)^2 \,\mathrm{d}p\right)^{1/2},$$

dove F_μ^{-1} è la funzione quantile della distribuzione spettrale.

Nota metodologica

Questa metrica permette di individuare layer "instabili" durante il fine-tuning (alta W_2) e layer "congelati" (bassa W_2), guidando strategie di layer-wise learning rate (LLRD).

1.7.4 §552 Riepilogo della Sezione A1

Strumento RMT	**Oggetto**	**Insight**
Legge semicircolare	Matrici casuali quadrate	Baseline “no-signal”
Legge Marchenko-Pastur	Covarianza campionaria	Soglia di rumore λ_+
Power-law spettrale	Pesi addestrati	Segnale di generalizzazione
Analisi SVD	Layer singoli	Rango effettivo, pruning
Distanza di Wasserstein	Evoluzione spettrale	Diagnostica fine-tuning

2 Sezione B — Topologia Applicata al Deep Learning

2.1 §568 Fondamenti di topologia algebrica

2.1.1 Complessi simpliciali

Definizione 2.1: Simplesso e complesso simpliciale

Un **k-simplesso** è l'inviluppo convesso di $k+1$ punti affini generici $\{v_0, v_1, \dots, v_k\} \subset \mathbb{R}^d$:

$$\sigma^k = \left\{ \sum_{i=0}^{k} \lambda_i v_i \;:\; \lambda_i \geq 0,\ \sum_{i=0}^{k} \lambda_i = 1 \right\}.$$

Casi: 0-simplesso = punto, 1-simplesso = spigolo, 2-simplesso = triangolo, 3-simplesso = tetraedro.
Un **complesso simpliciale** K è una collezione di simplessi chiusa per facce: se $\sigma \in K$ e $\tau \subseteq \sigma$, allora $\tau \in K$.

2.1.2 Omologia simpliciale

Definizione 2.2: Gruppo delle catene e operatore bordo

Il **gruppo delle k-catene** $C_k(K; \mathbb{R})$ è lo spazio vettoriale generato dai k-simplessi di K. L'**operatore bordo** $\partial_k : C_k \to C_{k-1}$ è definito linearmente su ogni k-simplesso:

$$\partial_k [v_0, v_1, \dots, v_k] = \sum_{i=0}^{k} (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_k],$$

dove $\hat{v}_i$ indica che v_i è omesso. La proprietà fondamentale è $\partial_{k-1} \circ \partial_k = 0$.

Definizione 2.3: Gruppo di omologia

Il **k-esimo gruppo di omologia** con coefficienti in $\mathbb{R}$ è

$$H_k(K; \mathbb{R}) \triangleq \ker(\partial_k) / \operatorname{im}(\partial_{k+1}).$$

La sua dimensione è il **k-esimo numero di Betti**:

$$\beta_k \triangleq \dim_{\mathbb{R}} H_k(K; \mathbb{R}).$$

Interpretazione: β_0 conta le componenti connesse, β_1 i buchi 1-dimensionali (anse), β_2 le cavità 2-dimensionali.

ESEMPIO NUMERICO — Numeri di Betti di forme geometriche semplici

Spazio	β_0	β_1	β_2
Punto	1	0	0
Cerchio S^1	1	1	0
Sfera S^2	1	0	1
Toro T^2	1	2	1
Due cerchi disgiunti	2	2	0

2.2 §569 Omologia persistente

L'omologia persistente è la risposta della topologia algebrica al seguente problema: dato un insieme di punti $\mathcal{X} = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ (per esempio, uno strato nascosto di una rete neurale), come si descrivono le sue caratteristiche topologiche in modo robusto al rumore?

2.2.1 Filtrazione di Vietoris-Rips

Definizione 2.4: Complesso di Vietoris-Rips

Per un insieme di punti $\mathcal{X}$ e un parametro di scala $\varepsilon \geq 0$, il **complesso di Vietoris-Rips** $\mathrm{VR}(\mathcal{X}, \varepsilon)$ è il complesso simpliciale i cui k-simplessi sono tutti i sottoinsiemi $\{x_{i_0}, \dots, x_{i_k}\}$ di diametro $\leq \varepsilon$:

$$\{x_{i_0}, \dots, x_{i_k}\} \in \mathrm{VR}(\mathcal{X}, \varepsilon) \iff \max_{a,b} d(x_{i_a}, x_{i_b}) \leq \varepsilon.$$

Al variare di ε, si ottiene una **filtrazione** $\mathrm{VR}(\mathcal{X}, \varepsilon_1) \subseteq \mathrm{VR}(\mathcal{X}, \varepsilon_2)$ per $\varepsilon_1 \leq \varepsilon_2$.

2.2.2 Diagrammi di persistenza

Man mano che ε cresce da 0 a ∞, le classi di omologia nascono (*birth*) e muoiono (*death*).

Definizione 2.5: Diagramma di persistenza

Il **diagramma di persistenza** $\mathrm{Dgm}_k(\mathcal{X})$ è il multiinsieme di coppie $(b_i, d_i) \in \mathbb{R}^2$ con $b_i \leq d_i$, dove b_i è il valore di ε alla nascita e d_i alla morte dell'i-esima classe di omologia di grado k. La **persistenza** di una feature è $\pi_i = d_i - b_i$.

Teorema 2.1: Stabilità dei diagrammi di persistenza (Cohen-Steiner et al., 2007)

Siano $\mathcal{X}, \mathcal{Y}$ due dataset. La distanza bottleneck tra i loro diagrammi di persistenza soddisfa

$$d_B(\mathrm{Dgm}_k(\mathcal{X}),\, \mathrm{Dgm}_k(\mathcal{Y})) \leq d_H(\mathcal{X}, \mathcal{Y}),$$

dove d_H è la distanza di Hausdorff e la distanza bottleneck è

$$d_B(P, Q) \triangleq \inf_{\gamma:\, P \to Q} \sup_{p \in P} \|p - \gamma(p)\|_\infty.$$

Dimostrazione. (Schema.) Si costruisce una corrispondenza γ tra le coppie (b, d) dei due diagrammi tramite il teorema dell'interleaving: due filtrazioni a distanza $\delta = d_H(\mathcal{X}, \mathcal{Y})$ sono δ-interlacciate, e l'δ-interleaving implica $d_B \leq \delta$. □ □

Il teorema di stabilità è cruciale: garantisce che piccole perturbazioni dei dati (rumore) producono piccole variazioni nei diagrammi di persistenza, rendendo la TDA robusta.

2.3 §570 Numeri di Betti del loss landscape

2.3.1 Costruzione della sublevel filtration

Sia $\mathcal{L} : \mathbb{R}^p \to \mathbb{R}$ la loss di una rete neurale in p parametri. Per ogni soglia $\ell \in \mathbb{R}$, si considera il **sottolivello**

$$\mathcal{L}^{-1}(-\infty, \ell] = \{\boldsymbol{\theta} \in \mathbb{R}^p : \mathcal{L}(\boldsymbol{\theta}) \leq \ell\}.$$

Al crescere di ℓ, questa regione si espande e la sua topologia cambia. Il diagramma di persistenza della filtrazione per sottolivelli cattura la struttura dei minimi, delle selle e dei massimi.

Proposizione 2.1: Corrispondenza Morse-topologia

Per una funzione di Morse generica $\mathcal{L}$ (gradiente non nullo fuori dai punti critici, Hessiana non singolare nei punti critici), c'è una corrispondenza biunivoca:

- Nascita di una componente connessa in $\mathcal{L}^{-1}(-\infty, \ell] \leftrightarrow$ Minimo locale di $\mathcal{L}$.
- Morte di un buco 1-dimensionale $\leftrightarrow$ Punto di sella con 1 direzione negativa (index-1 saddle).
- Morte di una componente connessa $\leftrightarrow$ Punto di sella che unisce due bacini.

2.3.2 Interpretazione per il training di reti neurali

ESEMPIO NUMERICO — Loss landscape di una rete a 2 layer

Si considera una rete fully-connected con $p = 1000$ parametri addestrata su MNIST (10 classi). Al variare della soglia ℓ:

1. A $\ell = \ell_{\min}$ (minimo globale), $\beta_0 = 1$: unica componente.
2. Al crescere di ℓ, compaiono minimi locali: β_0 aumenta a $\sim$ 30–50 componenti per ℓ pari a 1.2× il minimo.
3. I punti di sella di index-1 connettono questi bacini: β_1 cresce poi decresce.
4. Per $\ell \to \infty$ il loss landscape diventa connesso ($\beta_0 = 1$) e topologicamente banale.

La *persistenza media* dei minimi locali (distanza birth-to-death) è un proxy della profondità dei bacini: alta persistenza indica minimi "a gradino" (cattivi per la generalizzazione), bassa persistenza indica basin flat (buoni).

2.4 §571 L'algoritmo Mapper

L'algoritmo Mapper (Singh, Mémoli, Carlsson, 2007) è uno strumento per la visualizzazione e l'analisi topologica di spazi ad alta dimensione tramite un grafo riassuntivo.

Definizione 2.6: Algoritmo Mapper

Dati un insieme di punti $\mathcal{X}$, una funzione di filtro $f : \mathcal{X} \to \mathbb{R}^k$ (tipicamente $k = 1, 2$), e una copertura $\mathcal{U} = \{U_i\}$ di $f(\mathcal{X})$, l'algoritmo Mapper costruisce un grafo G_{Mapper} come segue:

1. Per ogni $U_i \in \mathcal{U}$, considera il preimage $f^{-1}(U_i) \cap \mathcal{X}$.
2. Esegui il clustering (es. single-linkage) su ogni preimage per ottenere cluster C_{ij}.
3. Ogni cluster C_{ij} diventa un nodo del grafo.
4. Due nodi C_{ij} e $C_{i'j'}$ sono connessi da uno spigolo se $C_{ij} \cap C_{i'j'} \neq \emptyset$.

Il grafo Mapper è un *grafo nervio* della copertura risultante dal clustering, e fornisce un'approssimazione topologica dello spazio dei dati.

2.4.1 Mapper sulle rappresentazioni neurali

Proposizione 2.2: Convergenza del Mapper nel limite continuo

Quando la copertura $\mathcal{U}$ si raffina (overlap $\to 0$, diametro degli insiemi $\to 0$), il complesso nervio del Mapper converge (in distanza di Gromov-Hausdorff) al Reeb graph della funzione f sullo spazio $\mathcal{X}$. Il Reeb graph cattura la topologia globale: le sue componenti connesse corrispondono a β_0, i suoi cicli a β_1.

ESEMPIO NUMERICO — Mapper su BERT: struttura del contesto linguistico

Applicando Mapper alle rappresentazioni del layer 12 di BERT con funzione di filtro $f =$ prima componente UMAP:

- Il grafo Mapper mostra 3 cluster principali ($\beta_0 = 3$) corrispondenti a costruzioni sintattiche diverse (SVO, SV, VO).
- Due loop ($\beta_1 = 2$) corrispondono a costruzioni ambigue che si connettono a più cluster.
- I layer profondi (10–12) hanno β_0 inferiore ($\sim$ 1–2) rispetto ai layer superficiali (1–3, $\beta_0 \sim$ 8–12), indicando una progressiva integrazione dell'informazione.

2.5 §572 Distanza di Wasserstein tra diagrammi di persistenza

Per confrontare topologicamente due reti neurali o due fasi di training, si utilizza la *distanza di Wasserstein* tra diagrammi di persistenza, più stabile della bottleneck per outlier:

Definizione 2.7: Distanza di Wasserstein tra diagrammi

La **distanza di Wasserstein di ordine** q tra due diagrammi P e Q è

$$W_q(P,Q) = \left(\inf_{\gamma:\, P\cup\Delta \to Q\cup\Delta} \sum_{p\in P\cup\Delta} \|p - \gamma(p)\|_\infty^q \right)^{1/q},$$

dove Δ è la diagonale (feature di persistenza zero), che permette di assegnare punti non accoppiati alla diagonale con costo pari alla metà della loro persistenza.

2.6 §573–§577 Betti del loss landscape nei Transformer

2.6.1 §573 Topologia delle rappresentazioni di layer

Sia $\phi^{(\ell)}(\mathcal{X}) = \{h_i^{(\ell)}\}_{i=1}^n$ il dataset proiettato al layer ℓ. I numeri di Betti $\beta_k(\mathrm{VR}(\phi^{(\ell)}(\mathcal{X}), \varepsilon))$ come funzione di ℓ e ε forniscono una **firma topologica** del layer.

Proposizione 2.3: Collasso topologico per layer profondi

Per reti MLP profonde addestrate su dati strutturati:

1. β_1 delle rappresentazioni di layer superficiali è significativamente più alto di zero (struttura complessa).
2. β_1 decresce monotonicamente con la profondità del layer.
3. Al layer di output, $\beta_0 = C$ (numero di classi) e $\beta_k = 0$ per $k \geq 1$: le classi sono cluster isolati e convessi.

Questo è il **collasso topologico**: la rete "semplifica" progressivamente la topologia dei dati fino a separarla linearmente.

2.6.2 §574 Persistent homology del training

Tracciare l'evoluzione dei diagrammi di persistenza durante il training rivela fasi distinte:

1. **Fase 1 (early training, $t < T_1$):** Aumento di β_1 (emergenza di strutture toroidali nello spazio dei pesi).
2. **Fase 2 (mid training, $T_1 < t < T_2$):** β_1 decresce; le anse si chiudono; la rete comprime le rappresentazioni.
3. **Fase 3 (late training, $t > T_2$):** β_0 si stabilizza; le componenti rimaste sono stabili.

Questo pattern è coerente con l'Information Bottleneck (Tishby et al.) trattato nel Volume II (§90): la fase di compressione corrisponde alla riduzione di β_1.

2.6.3 §575 Topologia e generalizzazione

Proposizione 2.4: Correlazione Betti-generalizzazione

Sia $\bar{\beta}_1$ la persistenza media dei feature di grado 1 nel diagramma di persistenza delle rappresentazioni del penultimo layer. Empiricamente (Guss & Salakhutdinov, 2018; Naitzat et al., 2020):

$$\text{generalization gap} \propto \bar{\beta}_1(\phi^{(L-1)}),$$

ovvero reti con rappresentazioni topologicamente più semplici generalizzano meglio.

2.6.4 §576 Calcolo efficiente dell'omologia persistente

L'algoritmo standard per il calcolo dell'omologia persistente è la **riduzione della matrice dei bordi** (Edelsbrunner & Harer, 2010):

[Riduzione della matrice dei bordi]

Require: Matrice dei bordi $\mathbf{D}$ (matrice della filtrazione)

Inizializza $\mathbf{R} \leftarrow \mathbf{D}$, $\mathbf{V} \leftarrow \mathbf{I}$
for $j = 1$ to m **do** ▷ m = numero di simplessi
  while $\exists\, i < j : \text{low}(\mathbf{R}_{:,i}) = \text{low}(\mathbf{R}_{:,j})$ **do**
    $\mathbf{R}_{:,j} \leftarrow \mathbf{R}_{:,j} + \mathbf{R}_{:,i}$ (somma su $\mathbb{F}_2$)
    $\mathbf{V}_{:,j} \leftarrow \mathbf{V}_{:,j} + \mathbf{V}_{:,i}$
  end while
end for
return coppie (b, d) dai pivot di $\mathbf{R}$

La complessità è $O(m^3)$ nel caso peggiore, dove m è il numero di simplessi. Implementazioni pratiche (Ripser, Gudhi) usano ottimizzazioni che riducono il costo a $O(m^2)$ in media.

2.6.5 §577 Riepilogo della Sezione B

Strumento TDA	**Applicazione**	**Insight**
Omologia persistente	Spazio dei dati	Buchi/componenti robusti
Diagramma di persistenza	Confronto dataset	Firma topologica
Numeri di Betti β_k	Loss landscape	Struttura dei minimi
Algoritmo Mapper	Rappresentazioni neurali	Grafo topologico
W_q tra diagrammi	Fasi del training	Evoluzione topologica

3 Sezione C — Active Learning e Bayesian Experimental Design

3.1 §588 Formulazione del problema di active learning

Nell'apprendimento supervisionato classico, si assume che il dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ sia dato. Nell'**active learning** il modello può scegliere *quali* esempi etichettare, minimizzando il costo di acquisizione.

Definizione 3.1: Setting di active learning

Sia:

- $\mathcal{U} = \{x_1, \dots, x_M\}$ il **pool non etichettato** (unlabeled pool);
- $\mathcal{L} = \{(x_i, y_i)\}_{i=1}^{n}$ il **labeled set** iniziale;
- $\mathcal{A} : \mathcal{U} \times \mathcal{M} \to \mathcal{U}$ la **strategia di acquisizione** (acquisition function) che seleziona il prossimo esempio x^* da etichettare;
- B il **budget** totale di etichettature.

Il ciclo di active learning ripete per $t = 1, \dots, B$:

1. Addestra il modello $\mathcal{M}_t$ su $\mathcal{L}$;
2. $x_t^* = \mathcal{A}(\mathcal{U}, \mathcal{M}_t)$;
3. Etichetta x_t^*: $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_t^*, y_t^*)\}$;
4. $\mathcal{U} \leftarrow \mathcal{U} \setminus \{x_t^*\}$.

3.2 §589 Uncertainty Sampling

L'uncertainty sampling (Lewis & Gale, 1994) seleziona il punto su cui il modello è più incerto. Per un classificatore $p(y|x, \theta)$ con C classi:

Definizione 3.2: Strategie di uncertainty sampling

1. **Least Confident (LC):**

$$x^* = \operatorname*{argmin}_{x \in \mathcal{U}} \max_{c} p(y = c \mid x, \theta),$$

   ovvero il punto con la massima probabilità più bassa.

2. **Margin Sampling:**

$$x^* = \operatorname*{argmin}_{x \in \mathcal{U}} \left[p(y = c_1 \mid x, \theta) - p(y = c_2 \mid x, \theta) \right],$$

   dove c_1, c_2 sono le prime due classi per probabilità.

3. **Entropy Sampling:**

$$x^* = \operatorname*{argmax}_{x \in \mathcal{U}} H[p(y \mid x, \theta)] = \operatorname*{argmax}_{x \in \mathcal{U}} - \sum_{c=1}^{C} p(y = c \mid x, \theta) \ln p(y = c \mid x, \theta).$$

Proposizione 3.1: Equivalenza per classificatori binari

Per $C = 2$, le tre strategie sono equivalenti in senso ordinale: se $p = p(y = 1|x, \theta)$, allora

$$\text{LC:} \quad \min(p, 1-p), \quad \text{Margin:} \quad |2p-1|, \quad \text{Entropy:} \quad -p\ln p - (1-p)\ln(1-p),$$

che sono tutte funzioni strettamente decrescenti di $|p - 0.5|$. Pertanto argmin su tutte e tre coincide.

ESEMPIO NUMERICO — Uncertainty sampling su classificazione di testi

Classificatore BERT fine-tuned su 100 esempi etichettati (sentiment analysis, 2 classi). Pool non etichettato: 10 000 esempi.

Strategia	Accuracy @200 label	Accuracy @500 label
Random sampling	82.3%	87.1%
Least Confident	85.1%	89.4%
Entropy	85.7%	89.8%

L'uncertainty sampling riduce il costo di etichettatura del $\sim 40\%$ per raggiungere la stessa accuracy del campionamento casuale.

3.3 §590 Query-by-Committee (QBC)

Il Query-by-Committee (Seung et al., 1992) mantiene un *comitato* di modelli $\{h_1, \dots, h_C\}$ e seleziona il punto su cui il comitato è più in disaccordo.

Definizione 3.3: Disaccordo del comitato

Il **vote entropy** di un punto x rispetto al comitato è

$$\mathrm{VE}(x) \triangleq -\sum_{c=1}^{C} \frac{V(y=c,x)}{C} \ln \frac{V(y=c,x)}{C},$$

dove $V(y = c, x) = |\{h_i : h_i(x) = c\}|$ è il numero di "voti" per la classe c. La strategia QBC seleziona

$$x^* = \operatorname*{argmax}_{x \in \mathcal{U}} \mathrm{VE}(x).$$

3.3.1 QBC bayesiano via approssimazione della posterior

Nella versione bayesiana, il comitato è un campione dalla posterior: $h_i \sim p(\theta|\mathcal{D})$. Il disaccordo diventa la **varianza di predizione**:

$$\sigma^2_{pred}(x) = \mathbb{E}_\theta\Big[\big(f(x;\theta) - \mathbb{E}_\theta[f(x;\theta)]\big)^2\Big].$$

Per reti neurali, l'approssimazione è realizzata con **MC Dropout** (Gal & Ghahramani, 2016): si esegue dropout in inferenza per T forward pass e si calcola la varianza empirica:

$$\hat{\sigma}^2_{pred}(x) = \frac{1}{T}\sum_{t=1}^{T} (\hat{y}_t - \bar{y})^2, \quad \bar{y} = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t.$$

Proposizione 3.2: MC Dropout come approssimazione variazionale

Sia $q_\phi(\theta)$ la variational distribution del dropout. Il predictor MC Dropout minimizza la divergenza KL:

$$\phi^* = \operatorname*{argmin}_\phi D_{\mathrm{KL}}(q_\phi(\theta) \,\|\, p(\theta|\mathcal{D}))\,.$$

La varianza MC Dropout è quindi una stima della varianza *epistémica* (dovuta all'incertezza sul modello), separata dalla varianza *aleatoria* (rumore intrinseco dei dati).

3.4 §591 Expected Information Gain (EIG)

L'**Expected Information Gain** (EIG) è la misura bayesiana ottimale per l'active learning: seleziona il punto che massimizza la riduzione attesa dell'incertezza sui parametri.

Definizione 3.4: Expected Information Gain

$$\mathrm{EIG}(x) \triangleq I(\theta; y \mid x, \mathcal{D}) = H[p(y \mid x, \mathcal{D})] - \mathbb{E}_\theta[H[p(y \mid x, \theta)]]\,,$$

dove $I(\theta; y \mid x, \mathcal{D})$ è la **mutua informazione** tra i parametri θ e l'osservazione y al punto x.

Proposizione 3.3: EIG come mutua informazione

$\mathrm{EIG}(x)$ è equivalente alla mutua informazione $I(\theta; y|x, \mathcal{D}) \geq 0$, e vale zero se e solo se $y \perp\!\!\!\perp \theta \mid x, \mathcal{D}$, ovvero se l'osservazione y non dà alcuna informazione sui parametri. L'EIG è anche simmetrico:

$$I(\theta; y \mid x, \mathcal{D}) = I(y; \theta \mid x, \mathcal{D}) = H[\theta|\mathcal{D}] - H[\theta|y, x, \mathcal{D}].$$

3.4.1 BALD: Bayesian Active Learning by Disagreement

BALD (Houlsby et al., 2011) è la formulazione pratica dell'EIG:

$$\mathrm{BALD}(x) = H[\mathbb{E}_\theta[p(y|x,\theta)]] - \mathbb{E}_\theta[H[p(y|x,\theta)]]\,.$$

Il primo termine è l'entropia della predizione media (incertezza totale), il secondo è l'entropia media per parametri fissi (incertezza aleatoria). La loro differenza è l'incertezza epistemica.

ESEMPIO NUMERICO — Confronto quantitativo BALD vs Entropy

Su MNIST con 100 label iniziali e CNN + MC Dropout ($T = 50$):

Metodo	@200	@500	@1000
Random	91.2%	95.8%	97.6%
Entropy	93.1%	96.9%	98.1%
BALD	94.8%	97.5%	98.4%

BALD supera l'entropy sampling perché separa l'incertezza epistemica da quella aleatoria: non si spreca il budget su esempi etichettati con rumore intrinseco.

3.5 §592 Optimal Experimental Design bayesiano

L'Optimal Experimental Design (OED) è la generalizzazione dell'active learning al caso in cui si vuole ottimizzare un *criterio di utilità globale* per la raccolta dell'intero batch di osservazioni.

Definizione 3.5: Criteri OED classici

Per un modello lineare $y = \mathbf{X}\boldsymbol{\theta} + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, con informazione di Fisher $\mathbf{M}(\xi) = \int \boldsymbol{x}\boldsymbol{x}^\top \, \mathrm{d}\xi(\boldsymbol{x})$:

1. **D-ottimalità:** $\max_\xi \det(\mathbf{M}(\xi))$ (minimizza il volume dell'ellissoide di confidenza).
2. **A-ottimalità:** $\max_\xi \operatorname{tr}(\mathbf{M}(\xi)^{-1})^{-1}$ (minimizza la varianza media degli stimatori).
3. **G-ottimalità:** $\min_\xi \max_x d(x, \xi)$ (minimizza la massima leverage).
4. **E-ottimalità:** $\max_\xi \lambda_{\min}(\mathbf{M}(\xi))$ (massimizza il più piccolo autovalore dell'informazione).

Il **General Equivalence Theorem** (Kiefer & Wolfowitz, 1960) garantisce che D- e G-ottimalità coincidono.

Teorema 3.1: General Equivalence Theorem

Sia ξ^* un design D-ottimale per il modello lineare con p parametri. Allora ξ^* è anche G-ottimale, e vale:

$$\max_{x \in \mathcal{X}} d(x, \xi^*) = p \quad \text{(pari al numero di parametri)},$$

dove $d(x, \xi) = \boldsymbol{x}^\top \mathbf{M}(\xi)^{-1} \boldsymbol{x}$ è la *standardized variance of prediction*.

Dimostrazione. (Outline.) Si dimostra per convessità del criterio D ($\log\det$) che ξ^* soddisfa le condizioni KKT equivalenti a $\max_x d(x, \xi^*) = p$. L'equivalenza con la G-ottimalità segue da questa condizione. □ □

3.5.1 OED bayesiano non lineare

Per modelli non lineari (reti neurali), l'informazione di Fisher $\mathbf{F}(\theta)$ dipende dai parametri, e si usa l'approssimazione di Laplace della posterior. La **D-ottimalità bayesiana** corrisponde a massimizzare il *log-determinant* della posterior precision:

$$x^* = \operatorname*{argmax}_{x \in \mathcal{U}} \ln\det(\boldsymbol{\Lambda}_{prior} + \mathbf{F}(\theta, x)) \,,$$

dove $\mathbf{F}(\theta, x) = -\mathbb{E}_y[\nabla^2_\theta \ln p(y|x, \theta)]$ è la contribuzione di (x, y) all'informazione di Fisher.

3.6 §593 Connessione con i Gaussian Process

Il legame fondamentale tra active learning e Gaussian Process (trattati nel Volume V) è che i GP forniscono un framework analitico dove l'EIG è calcolabile in forma chiusa.

Sia $f \sim \mathcal{GP}(\mu, k)$ con kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Dati $\mathcal{D} = (\mathbf{X}, \boldsymbol{y})$, la posterior è:

Proposizione 3.4: Posterior GP e EIG analitico

La posterior di un GP su n osservazioni è ancora un GP:

$$\mu_n(x) = \mu(x) + \boldsymbol{k}_n(x)^\top (\mathbf{K}_n + \sigma^2\mathbf{I})^{-1}(\boldsymbol{y} - \mu(\mathbf{X})),$$
$$k_n(x, x') = k(x, x') - \boldsymbol{k}_n(x)^\top (\mathbf{K}_n + \sigma^2\mathbf{I})^{-1}\boldsymbol{k}_n(x'),$$

dove $[\boldsymbol{k}_n(x)]_i = k(x, x_i)$ e $[\mathbf{K}_n]_{ij} = k(x_i, x_j)$.

L'**EIG** per aggiungere il punto x è

$$\mathrm{EIG}_{GP}(x) = \frac{1}{2}\ln\left(1 + \frac{k_n(x,x)}{\sigma^2}\right) = \frac{1}{2}\ln\left(1 + \frac{\sigma_n^2(x)}{\sigma^2}\right),$$

dove $\sigma_n^2(x) = k_n(x,x)$ è la **varianza predittiva**.

Dimostrazione. $I(\theta; y|x, \mathcal{D}) = H[p(y|x,\mathcal{D})] - H[p(y|x, f(x))]$ dove entrambi i termini sono gaussiani. Poiché $p(y|x,\mathcal{D}) = \mathcal{N}(\mu_n(x), \sigma_n^2(x)+\sigma^2)$ e $H[\mathcal{N}(\mu,\sigma^2)] = \frac{1}{2}\ln(2\pi e\sigma^2)$, la differenza è $\frac{1}{2}\ln\frac{\sigma_n^2(x)+\sigma^2}{\sigma^2}$. □ □

Quindi per i GP, l'**active learning ottimale** corrisponde a scegliere il punto con la massima varianza predittiva:

$$x^* = \operatorname*{argmax}_{x\in\mathcal{U}} \sigma_n^2(x).$$

Questo è anche la strategia greedy per la **massimizzazione della mutua informazione** nel caso GP.

ESEMPIO NUMERICO — Active GP regression su dati di sensori

Dataset: 500 misurazioni di temperatura in una stanza ($\mathcal{X} = [0,1]^2$, coordinate spaziali). Budget: 50 misurazioni selettive. Kernel: RBF con $\ell = 0.1$, $\sigma_f = 1$.

- Random sampling @50: RMSE $= 0.142 \pm 0.023$.
- Max-variance GP @50: RMSE $= 0.067 \pm 0.011$.

L'active learning riduce il RMSE del 53% con lo stesso budget.

3.7 §594 Active learning su sequenze e LLM

L'active learning per LLM presenta sfide specifiche:

1. Lo spazio di input $\mathcal{U}$ è discreto (tokens).
2. Il calcolo dell'EIG richiede forward pass multipli.
3. La distribuzione delle label è strutturata (istruzioni, risposte).

3.7.1 EIG per instruction tuning

Nel contesto del fine-tuning di LLM tramite RLHF/SFT, il pool $\mathcal{U}$ è un insieme di prompt. L'EIG seleziona i prompt che massimizzano la riduzione di incertezza sui parametri del modello. Una proxy efficiente è l'**EL2N score** (Error L2-Norm):

$$\mathrm{EL2N}(x) = \mathbb{E}_y\left[\|\hat{p}(y|x) - p(y|x)\|_2^2\right] \approx \frac{1}{T}\sum_{t=1}^{T}\|\hat{p}_t(y|x) - \bar{p}(y|x)\|_2^2,$$

calcolato su T checkpoint del training, che approssima la varianza di predizione senza richiedere l'accesso alla posterior.

3.8 §595 Batch active learning e submodularità

In pratica si seleziona un *batch* $\mathcal{B}$ di b esempi da etichettare simultaneamente, non uno alla volta.

Definizione 3.6: Submodularità

Una funzione $F : 2^{\mathcal{U}} \to \mathbb{R}$ è **submodulare** se, per ogni $A \subseteq B \subseteq \mathcal{U}$ e ogni $x \notin B$:

$$F(A \cup \{x\}) - F(A) \geq F(B \cup \{x\}) - F(B).$$

Equivalentemente, F ha *rendimenti marginali decrescenti*.

Teorema 3.2: Approssimazione greedy submodulare (Nemhauser et al., 1978)

Sia F submodulare e monotona non decrescente con $F(\emptyset) = 0$. L'algoritmo greedy che seleziona iterativamente $x_t = \operatorname{argmax}_{x \notin \mathcal{B}_{t-1}} F(\mathcal{B}_{t-1} \cup \{x\})$ garantisce

$$F(\mathcal{B}_{greedy}) \geq \left(1 - \frac{1}{e}\right) F(\mathcal{B}^*),$$

dove $\mathcal{B}^*$ è il batch ottimale di cardinalità b.

Proposizione 3.5: Submodularità della mutua informazione GP

La funzione $F(\mathcal{B}) = I(f_{\mathcal{B}}; \boldsymbol{y}_{\mathcal{B}})$ (mutua informazione tra i valori del GP su $\mathcal{B}$ e le relative osservazioni rumorose) è submodulare e monotona. Pertanto, la selezione greedy di batch massimizza la mutua informazione con garanzia $(1 - 1/e)$.

Dimostrazione. La submodularità segue dalla proprietà della varianza condizionale dei GP: $\sigma^2(x|\mathcal{A}) \geq \sigma^2(x|\mathcal{B})$ per $\mathcal{A} \subseteq \mathcal{B}$, che implica rendimenti marginali decrescenti per l'EIG. □ □

3.9 §596–§598 Core-Set e diversità

3.9.1 §596 Il metodo Core-Set

Il metodo Core-Set (Sener & Savarese, 2018) seleziona il batch $\mathcal{B}$ come soluzione al k-center problem:

$$\mathcal{B}^* = \operatorname*{argmin}_{\mathcal{B} \subseteq \mathcal{U},\, |\mathcal{B}| = b} \max_{x \in \mathcal{U}} \min_{x' \in \mathcal{B} \cup \mathcal{L}} d(x, x'),$$

dove d è tipicamente la distanza euclidea nello spazio delle feature dell'ultimo layer.

Proposizione 3.6: Bound di generalizzazione del Core-Set

Sia $\delta^* = \max_x \min_{x' \in \mathcal{B} \cup \mathcal{L}} d(x, x')$ il *covering radius*. Allora per un modello Lipschitz con costante L_f:

$$\mathcal{L}_{test}(f_{\mathcal{L}}) \leq \mathcal{L}_{train}(f_{\mathcal{L}}) + L_f\, \delta^* + O\left(\sqrt{\frac{\ln(1/\delta)}{|\mathcal{L}|}}\right),$$

con probabilità $1 - \delta$. Minimizzare δ^* (Core-Set) minimizza direttamente questo bound di generalizzazione.

3.9.2 §597 PowerBALD: batch EIG con diversità

PowerBALD (Kirsch et al., 2021) combina EIG e diversità selezionando il primo esempio con BALD e i successivi campionando proporzionalmente all'EIG (con temperatura τ):

$$p(x_t^* = x) \propto \mathrm{BALD}(x)^{1/\tau},$$

evitando la ridondanza tipica dell'EIG greedy che seleziona cluster di esempi simili.

3.9.3 §598 Confronto sistematico delle strategie

Metodo	Tipo	Bayesiano	Batch	Complessità
Random	Baseline	No	Sì	$O(1)$
Least Confident	Uncertainty	No	Sì	$O(M)$
Entropy	Uncertainty	Parz.	Sì	$O(M)$
QBC / MC Dropout	Comitato	Sì	Sì	$O(TM)$
BALD	EIG bayesiano	Sì	No	$O(TM)$
PowerBALD	EIG + diversità	Sì	Sì	$O(TM)$
Core-Set	Geometrico	No	Sì	$O(M \log M)$
GP max-var	EIG analitico	Sì	Sì	$O(n^3 + Mn^2)$

3.10 §599 Connessione con il Bayesian Experimental Design per reti neurali profonde

Il **Bayesian Experimental Design** (BED) generalizza l'OED al caso in cui il modello è una rete neurale profonda con posterior approssimata.

Definizione 3.7: BED con approssimazione di Laplace

Sia $p(\theta|\mathcal{D}) \approx \mathcal{N}(\hat{\theta}, \mathbf{H}^{-1})$ l'approssimazione di Laplace della posterior, dove $\hat{\theta}$ è il MAP e $\mathbf{H} = -\nabla^2_\theta \ln p(\theta|\mathcal{D})|_{\hat{\theta}}$ è la Hessiana negativa. Il criterio di D-ottimalità bayesiana per selezionare x^* è:

$$x^* = \operatorname*{argmax}_{x \in \mathcal{U}} \ln\det\Big(\mathbf{H} + \mathbf{F}(\hat{\theta}, x)\Big) - \ln\det(\mathbf{H}),$$

dove $\mathbf{F}(\hat{\theta}, x) = \nabla_\theta f(x;\hat{\theta}) \nabla_\theta f(x;\hat{\theta})^\top$ è il **Fisher empirico** (rango 1) per l'osservazione x. Per il determinante di una perturbazione di rango 1 vale la formula di matrix determinant lemma:

$$\ln\det(\mathbf{H} + \boldsymbol{g}\boldsymbol{g}^\top) = \ln\det(\mathbf{H}) + \ln(1 + \boldsymbol{g}^\top \mathbf{H}^{-1} \boldsymbol{g}),$$

quindi la strategia BED-Laplace seleziona

$$x^* = \operatorname*{argmax}_{x \in \mathcal{U}} \boldsymbol{g}(x)^\top \mathbf{H}^{-1} \boldsymbol{g}(x), \quad \boldsymbol{g}(x) \triangleq \nabla_\theta f(x;\hat{\theta}).$$

Nota metodologica

La quantità $\boldsymbol{g}(x)^\top \mathbf{H}^{-1} \boldsymbol{g}(x)$ è la **influence function** dell'esempio x sui parametri, già introdotta nel Volume VII nel contesto dell'interpretabilità (§420). Il BED usa la stessa quantità per scegliere dove raccogliere i dati, completando un cerchio concettuale con l'interpretabilità.

3.11 §600 Teorema di no-free-lunch per l'active learning

Teorema 3.3: No-Free-Lunch per l'active learning (Dasgupta, 2005)

Per ogni strategia di active learning $\mathcal{A}$ e ogni $m \geq 1$, esiste una distribuzione $\mathcal{D}$ tale che, per apprendere un concetto da $\mathcal{H}$ con accuratezza $1 - \varepsilon$ e confidenza $1 - \delta$, $\mathcal{A}$ su $\mathcal{D}$ richiede almeno tante etichette quanto il campionamento passivo.

Dimostrazione. (Sketch.) Si costruisce una distribuzione avversariale che concentra tutta la massa informativa su punti non selezionati dalla strategia $\mathcal{A}$. Poiché $\mathcal{A}$ non può osservare le label non richieste, non ottiene informazione su quelle regioni, e richiede $\Omega(1/\varepsilon)$ etichette come il campionamento passivo. □ □

Il teorema NFL implica che l'active learning è utile solo con assunzioni sulla struttura della distribuzione dei dati o dello spazio delle ipotesi.

3.12 §601 Complessità del campionamento e legami con la VC theory

Proposizione 3.7: Sample complexity dell'active learning

Per un ipotesi class $\mathcal{H}$ con VC dimension d, la complessità di etichettatura dell'active learning è:

$$m_{active}(\varepsilon, \delta) = O\left(\frac{d}{\varepsilon} + \frac{\ln(1/\delta)}{\varepsilon}\right),$$

contro la complessità passiva

$$m_{passive}(\varepsilon, \delta) = O\left(\frac{d \ln(1/\varepsilon) + \ln(1/\delta)}{\varepsilon}\right).$$

Il guadagno del fattore $\ln(1/\varepsilon)$ è ottenibile solo per $\mathcal{H}$ *agreeably realizzabili* (Balcan et al., 2006).

3.13 §602 Riepilogo della Sezione C e del ciclo

Strumento	**Contesto**	**Garanzia**
Uncertainty sampling	Qualsiasi classificatore	Nessuna (euristico)
QBC / MC Dropout	Reti neurali bayesiane	Appross. posterior
BALD	Modelli con posterior	EIG ottimale greedy
GP max-var	Gaussian Process	EIG analitico esatto
Core-Set	Qualsiasi embedding	Bound di copertura
PowerBALD	Batch bayesiano	EIG + diversità
BED-Laplace	Reti neurali profonde	D-ottimalità approx.

La trattazione di questo volume chiude il cerchio del ciclo *Matematica delle Reti Neurali*: partendo dalle fondamenta dell'algebra lineare e del calcolo differenziale (Vol. I), passando per tutte le architetture moderne (Vol. II–VIII), per l'AI agentiva e generativa (Vol. IX), il ciclo approda qui alle radici *fisico-statistiche*, *topologiche* e *decisionali* del machine learning.

Questi tre filoni, apparentemente lontani, convergono in un'unica visione: l'apprendimento automatico è **inferenza geometrica in spazi ad alta dimensione**, governata da principi variazionali (minimizzazione della free energy, massimizzazione dell'informazione) e descritta da invarianti topologici (numeri di Betti, diagrammi di persistenza) che ne catturano la struttura essenziale indipendentemente dai dettagli architetturali.

A Appendice A — Glossario dei Simboli Principali

Simbolo	Significato
$\rho_{sc}(\lambda)$	Densità spettrale semicircolare di Wigner
$\rho_{MP}(\lambda)$	Densità di Marchenko-Pastur
$\lambda_{\pm}$	Bordi dello spettro Marchenko-Pastur
$\gamma = n/p$	Aspect ratio della matrice di covarianza
$\hat{\alpha}$	Esponente power-law dello spettro dei pesi
$\texttt{sr}(\mathbf{W})$	Stable rank di $\mathbf{W}$
$\texttt{erank}(\mathbf{W})$	Rango effettivo di $\mathbf{W}$
$q, q(x)$	Parametro d'ordine (RS) e funzione di Parisi (RSB)
$V(\kappa, \alpha)$	Volume di Gardner
$H_{SK}(\boldsymbol{\sigma})$	Hamiltoniana di Sherrington-Kirkpatrick
K	Complesso simpliciale
∂_k	Operatore bordo di grado k
$H_k(K; \mathbb{R})$	k-esimo gruppo di omologia (coefficienti reali)
β_k	k-esimo numero di Betti
$\mathrm{VR}(\mathcal{X}, \varepsilon)$	Complesso di Vietoris-Rips
$\mathrm{Dgm}_k(\mathcal{X})$	Diagramma di persistenza di grado k
(b_i, d_i)	Coppia (birth, death) di una feature persistente
$\pi_i = d_i - b_i$	Persistenza della i-esima feature
$d_B(P, Q)$	Distanza bottleneck tra diagrammi
$W_q(P, Q)$	Distanza di Wasserstein di ordine q tra diagrammi
G_{Mapper}	Grafo prodotto dall'algoritmo Mapper
$\mathcal{A}$	Strategia di acquisizione (active learning)
$\mathcal{U}$	Pool non etichettato
$\mathrm{EIG}(x)$	Expected Information Gain al punto x
$\mathrm{BALD}(x)$	Bayesian Active Learning by Disagreement
$\mathbf{M}(\xi)$	Matrice di informazione di Fisher del design ξ
$\sigma_n^2(x)$	Varianza predittiva del GP dopo n osservazioni
$\mathbf{H}$	Hessiana negativa della log-posterior (Laplace)
$\boldsymbol{g}(x)$	Gradiente del modello rispetto ai parametri in x

B Appendice B — Raccolta dei Teoremi Principali

1. **Legge semicircolare di Wigner** (§1.1): La distribuzione spettrale empirica di una matrice di Wigner GOE converge alla densità semicircolare su $[-2\sigma, 2\sigma]$.
2. **Legge di Marchenko-Pastur** (§1.2): La distribuzione spettrale empirica di $\mathbf{X}\mathbf{X}^\top/p$ per matrici casuali $\mathbf{X}$ converge alla legge MP con bordi $\lambda_\pm = \sigma^2(1 \pm \sqrt{\gamma})^2$.
3. **Free energy replica-symmetric** (§1.3): Formula chiusa per la free energy media del perceptron sferico nel'ansatz RS via extremizzazione in $q, \hat{q}$.
4. **Free energy di Parisi per il modello SK** (§1.4): La soluzione esatta (dimostrata da Talagrand 2006) è data dalla funzione $q(x)$ soddisfacente l'equazione di Parisi.
5. **Stabilità dei diagrammi di persistenza** (§2.1): $d_B(\mathrm{Dgm}_k(\mathcal{X}), \mathrm{Dgm}_k(\mathcal{Y})) \leq d_H(\mathcal{X}, \mathcal{Y})$.
6. **General Equivalence Theorem** (§3.1): D-ottimalità e G-ottimalità coincidono; al design ottimale il massimo leverage è esattamente p.
7. **EIG analitico per GP** (§3.4): $\mathrm{EIG}_{GP}(x) = \frac{1}{2}\ln(1 + \sigma_n^2(x)/\sigma^2)$.
8. **Approssimazione greedy submodulare** (§3.2): Il greedy garantisce un fattore $(1 - 1/e)$ dell'ottimale per funzioni submodulari monotone.
9. **Submodularità della MI del GP** (§3.5): La funzione $F(\mathcal{B}) = I(f_\mathcal{B}; \boldsymbol{y}_\mathcal{B})$ è submodulare, garantendo l'applicabilità del greedy con ratio $(1 - 1/e)$.
10. **No-Free-Lunch per l'active learning** (§3.3): Non esiste una strategia universalmente superiore al campionamento passivo su tutte le distribuzioni.

C Appendice C — Note bibliografiche

Fisica statistica e reti neurali. La legge semicircolare è dovuta a Wigner (1955); la legge di Marchenko-Pastur a Marchenko e Pastur (1967). Il metodo delle repliche è stato introdotto da Edwards e Anderson (1975) e formalizzato da Parisi (1979, 1983). La prova rigorosa della formula di Parisi per il modello SK è di Talagrand (2006), con contributi cruciali di Guerra e Toninelli (2002). Il modello di Gardner (1988) estende l'analisi RS al perceptron sferico. Il fenomeno Heavy-Tail Self-Regularization è descritto da Martin e Mahoney (2021, *JMLR*).

Topologia applicata. L'omologia persistente è stata sistematizzata da Edelsbrunner, Letscher e Zomorodian (2002) e da Zomorodian e Carlsson (2005). Il teorema di stabilità è di Cohen-Steiner, Edelsbrunner e Harer (2007). L'algoritmo Mapper è di Singh, Mémoli e Carlsson (2007). Le applicazioni al loss landscape neurale si trovano in Guss e Salakhutdinov (2018) e Naitzat, Zhitnikov e Lim (2020).

Active learning e OED. L'uncertainty sampling è di Lewis e Gale (1994); il QBC di Seung, Opper e Sompolinsky (1992). BALD è di Houlsby et al. (2011). La teoria dell'OED ha radici in Kiefer e Wolfowitz (1960); il General Equivalence Theorem è classicamente attribuito a Whittle (1973). MC Dropout come approssimazione bayesiana è di Gal e Ghahramani (2016). Core-Set è di Sener e Savarese (2018); PowerBALD di Kirsch et al. (2021). Il teorema NFL per l'active learning è di Dasgupta (2005). La connessione tra BED e influence functions è discussa in Bickford Smith et al. (2023).

Nota conclusiva

Con il Volume X si chiude il ciclo *Matematica delle Reti Neurali*. I dieci volumi coprono, con rigore uniforme e derivazioni esplicite, la quasi totalità del deep learning classico e moderno: dai fondamenti dell'algebra lineare alle reti neurali profonde, dai modelli generativi all'allineamento, dalla biologia computazionale all'AI agentiva, fino alle connessioni più profonde con la fisica statistica, la topologia e la statistica decisionale.

Il ciclo è, probabilmente, l'opera più sistematica in lingua italiana sull'argomento. La speranza è che possa essere un riferimento duraturo per studenti, ricercatori e ingegneri che vogliono non solo *usare* il deep learning, ma comprenderne le fondamenta matematiche con la stessa profondità con cui i fisici e i matematici le hanno costruite.

Silvano Salvador
2025–2026

D Sezione A (continuazione) — Free Probability, Equazioni TAP e RFOT

D.1 §553 Free Probability Theory

La **Free Probability Theory** (Voiculescu, 1985) è l'analogo non-commutativo della teoria delle probabilità classica, sviluppato per trattare somme e prodotti di grandi matrici casuali con la stessa naturalezza con cui la teoria classica tratta somme di variabili scalari.

D.1.1 Libera indipendenza

Definizione D.1: Libera indipendenza

Siano $\mathbf{A}$ e $\mathbf{B}$ matrici casuali $n \times n$, e sia $\tau_n(\mathbf{M}) = \frac{1}{n}\operatorname{tr}(\mathbf{M})$ la traccia normalizzata. $\mathbf{A}$ e $\mathbf{B}$ sono **asintoticamente libere** se, per ogni polinomio non commutativo $p(a, b)$ privo di termine costante nelle variabili centrate $a = \mathbf{A} - \tau_n(\mathbf{A})\mathbf{I}$, $b = \mathbf{B} - \tau_n(\mathbf{B})\mathbf{I}$:

$$\tau_n(p(\mathbf{A}, \mathbf{B})) \to 0 \quad \text{per } n \to \infty.$$

La libera indipendenza è la generalizzazione non-commutativa dell'indipendenza classica: vale tra due matrici casuali "generiche" (es. GOE ruotate con rotazioni ortogonali indipendenti).

D.1.2 Trasformata R e addizione libera

Definizione D.2: Trasformata R

Data la trasformata di Stieltjes $G_\mu(z) = \int (\lambda - z)^{-1}\, \mathrm{d}\mu(\lambda)$ di una misura spettrale μ, si definisce:

$$K_\mu(z) = G_\mu^{-1}(z), \qquad R_\mu(z) = K_\mu(z) - \frac{1}{z}.$$

La **trasformata R** $R_\mu(z)$ è l'analogo non-commutativo del logaritmo della funzione caratteristica (cumulant transform).

Teorema D.1: Linearità della trasformata R (Voiculescu, 1986)

Se $\mathbf{A}$ e $\mathbf{B}$ sono asintoticamente libere con misure spettrali μ_A e μ_B, la misura spettrale di $\mathbf{A} + \mathbf{B}$ soddisfa la **additività libera**:

$$R_{\mu_A + \mu_B}(z) = R_{\mu_A}(z) + R_{\mu_B}(z).$$

Questo è l'analogo non-commutativo del teorema sulla somma di indipendenti (addizione dei cumulants): $\kappa_n(X+Y) = \kappa_n(X) + \kappa_n(Y)$.

D.1.3 Trasformata S e moltiplicazione libera

Proposizione D.1: Trasformata S e prodotto libero

Per il prodotto **AB** di matrici asintoticamente libere, la **trasformata S** (Voiculescu, 1987)

$$S_\mu(z) = \frac{1+z}{z\,\psi_\mu^{-1}(z)}, \qquad \psi_\mu(z) = \int \frac{\lambda z}{1-\lambda z}\,\mathrm{d}\mu(\lambda),$$

è **moltiplicativa**:

$$S_{\mu_{AB}}(z) = S_{\mu_A}(z)\cdot S_{\mu_B}(z).$$

ESEMPIO NUMERICO — Spettro di $\mathbf{W}^\top\mathbf{W}$ con free probability

Sia $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ una matrice di peso addestrata, dove $\mathbf{\Sigma}$ è il tensore dei valori singolari. Se la matrice di proiezione d'uscita $\mathbf{V}$ è "generica" rispetto agli input, la misura spettrale di $\mathbf{W}^\top\mathbf{W}$ è la *convoluzione libera* di μ_{Σ^2} (misura dei valori singolari al quadrato) con la legge di Marchenko-Pastur (dovuta al rumore di campionamento). Usando la trasformata S, la misura risultante si calcola analiticamente senza simulazione.

D.2 §554 Equazioni di TAP e campo medio per reti neurali

Le **equazioni di Thouless-Anderson-Palmer** (TAP, 1977) sono le equazioni di campo medio esatte per il modello SK, che vanno oltre l'approssimazione naive di campo medio aggiungendo il *termine di reazione di Onsager*.

D.2.1 Campo medio naive e termine di Onsager

Il campo medio naive per lo spin i è

$$m_i^{MF} = \tanh\left(\beta \sum_{j\neq i} J_{ij} m_j\right).$$

Le equazioni TAP correggono questo con il termine di reazione:

Definizione D.3: Equazioni TAP

Le equazioni di Thouless-Anderson-Palmer per il modello SK sono:

$$m_i = \tanh\left(\beta \sum_{j\neq i} J_{ij} m_j - \beta^2 (1-q)\, m_i\right), \quad q = \frac{1}{N}\sum_i m_i^2,$$

dove $-\beta^2(1-q)\,m_i$ è il **termine di reazione di Onsager** che conta il fatto che il campo sul sito i include un contributo di "backaction" da i stesso mediato dai vicini.

D.2.2 Equazioni TAP come message passing

Le equazioni TAP sono equivalenti all'algoritmo di **Belief Propagation approssimato** (AMP, Approximate Message Passing) su grafi densi.

Proposizione D.2: AMP come TAP vettoriale

L'algoritmo AMP per il compressed sensing,

$$\boldsymbol{z}^t = \boldsymbol{y} - \mathbf{A}\hat{\boldsymbol{x}}^t + \frac{1}{m}\hat{\boldsymbol{x}}^t \cdot \langle \eta_t' \rangle,$$
$$\hat{\boldsymbol{x}}^{t+1} = \eta_{t+1}(\mathbf{A}^\top \boldsymbol{z}^t + \hat{\boldsymbol{x}}^t),$$

dove η_t è una funzione di de-noising e $\langle \eta_t' \rangle$ è la sua derivata media, è la versione vettoriale delle equazioni TAP. Il termine $\frac{1}{m}\hat{\boldsymbol{x}}^t\langle \eta_t' \rangle$ è il termine di reazione di Onsager.

Teorema D.2: State evolution dell'AMP (Bayati-Montanari, 2011)

Per $\mathbf{A}$ con voci i.i.d. $\sim \mathcal{N}(0, 1/m)$ e $m/n \to \delta$, l'AMP converge e la sua dinamica è descritta esattamente dalla **state evolution**:

$$\sigma_{t+1}^2 = \sigma_z^2 + \frac{1}{\delta}\mathbb{E}_{\xi,X}\left[(\eta_t(\sqrt{\tau_t}\,\xi + X) - X)^2\right],$$

dove $X \sim p_X$, $\xi \sim \mathcal{N}(0,1)$, $\tau_t = \sigma_t^2/\delta$. Questa equazione determina lo stato (σ_t^2) a ogni iterazione in modo deterministico, cioè AMP è analizzabile esattamente nel limite termodinamico.

D.3 §555 RFOT: Random First-Order Transition Theory

La **Random First-Order Transition** (RFOT) theory (Kirkpatrick, Thirumalai, Wolynes, 1987–1989) è la teoria del campo medio esatta per vetri strutturali. È rilevante per il deep learning perché predice la struttura delle fasi del loss landscape.

Definizione D.4: Configurazioni e overlap

Nel framework RFOT, lo spazio dei parametri $\boldsymbol{\theta} \in \mathbb{R}^p$ è pensato come analogo allo spazio delle configurazioni di un sistema fisico. L'**overlap** tra due configurazioni è

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{1}{p}\boldsymbol{\theta} \cdot \boldsymbol{\theta}'.$$

La **funzione di distribuzione degli overlap** $P(Q)$ descrive la struttura dello spazio delle soluzioni: per $P(Q) = \delta(Q - Q^*)$ vi è un'unica struttura; per $P(Q)$ con supporto ampio vi è la RSB.

D.3.1 Transizioni di fase del loss landscape

La RFOT predice tre fasi qualitative nel loss landscape di una rete neurale al variare del carico $\alpha = P/N$:

Fase	Carico α	Struttura	GD behavior
Paramagnetica	$\alpha < \alpha_d$	Unico bacino, convesso	Convergenza rapida
Spin glass	$\alpha_d < \alpha < \alpha_s$	Molti bacini, RSB	Aging, metastabilità
Jamming	$\alpha > \alpha_s$	Nessuna soluzione	Divergenza

La transizione dinamica α_d è dove il gradiente discendente inizia a rallentare (critical slowing down); la transizione statica α_s è dove il sistema diventa infattibile.

Proposizione D.3: Critica lentezza e loss plateau

Nella fase spin glass ($\alpha_d < \alpha < \alpha_s$), la loss converge a un *plateau metastabile* $\mathcal{L}^*(\alpha) > \mathcal{L}_{min}$ con un tempo caratteristico che scala come:

$$t^*(\alpha) \sim \exp\left(\frac{A}{\alpha_s - \alpha}\right),$$

ovvero la convergenza al minimo globale diventa esponenzialmente lenta in prossimità della transizione statica. I plateau osservati durante il training di reti profonde su dataset saturi corrispondono a questa fenomenologia.

D.4 §556 Analisi spettrale della Hessiana

La **Hessiana** $\mathbf{H} = \nabla^2_\theta \mathcal{L} \in \mathbb{R}^{p\times p}$ della loss di una rete neurale contiene informazioni cruciali sulla geometria del loss landscape.

D.4.1 Decomposizione della Hessiana: GGN + residuo

Proposizione D.4: Decomposizione GGN

La Hessiana della loss MSE $\mathcal{L} = \frac{1}{n}\sum_i (f(x_i;\theta) - y_i)^2$ si decompone come:

$$\mathbf{H} = \underbrace{\frac{2}{n}\mathbf{J}^\top \mathbf{J}}_{GGN} + \underbrace{\frac{2}{n}\sum_i (f(x_i;\theta) - y_i)\nabla^2_\theta f(x_i;\theta)}_{R},$$

dove $\mathbf{J} = \nabla_\theta f(\mathbf{X};\theta) \in \mathbb{R}^{n\times p}$ è lo Jacobiano e $GGN = \frac{2}{n}\mathbf{J}^\top\mathbf{J}$ è la **Gauss-Newton matrix** (o Generalized GN). Al minimo ($R = 0$), $\mathbf{H} \approx GGN$: il GGN è una approssimazione della Hessiana valida vicino ai minimi.

D.4.2 Distribuzione spettrale della Hessiana via RMT

Proposizione D.5: Spettro bulk + outlier della Hessiana

Per una rete al termine del training, lo spettro di $\mathbf{H}$ mostra tipicamente:

1. Un **bulk** distribuito secondo la legge di Marchenko-Pastur (autovalori piccoli, legati alla curvatura del rumore).
2. Un numero piccolo di **outlier** (autovalori grandi, $O(\sqrt{n})$ rispetto al bulk), corrispondenti alle direzioni di curvatura informativa (le feature imparate).
3. Un numero di autovalori **esattamente zero** pari alla dimensione del kernel della Jacobiana $\mathbf{J}$, legato al grado di overparameterization.

ESEMPIO NUMERICO — Analisi spettrale su ResNet-50

Su ResNet-50 addestrata su ImageNet al 76% di top-1 accuracy:

- $\sim$ 25 autovalori outlier nel range $[10^2, 10^4]$ (corrispondenti approssimativamente ai $C = 1000$ cluster di classi, con ridondanza ridotta dalla decorrelazione dell'attenzione).
- Il bulk è ben approssimato dalla MP law con $\gamma \approx 0.3$.

- Circa $p - n \approx 23.5M$ autovalori zero (overparameterization: $p = 25M$ parametri, $n = 1.28M$ campioni).

D.5 §557–§567 Teoria mean-field per reti profonde

D.5.1 §557 Propagazione del segnale: teoria mean-field dei layer

La teoria mean-field per reti neurali profonde (Poole et al., 2016; Schoenholz et al., 2017) analizza come la varianza delle attivazioni si propaga layer per layer.

Definizione D.5: Mappa di varianza

Sia ϕ la funzione di attivazione e siano i pesi i.i.d. $W_{ij} \sim \mathcal{N}(0, \sigma_w^2/n)$, bias $b_i \sim \mathcal{N}(0, \sigma_b^2)$. La **mappa di varianza** $q^* \mapsto \mathcal{V}(q^*)$ è

$$\mathcal{V}(q) = \sigma_w^2 \mathbb{E}_{z\sim\mathcal{N}(0,q)}\left[\phi(z)^2\right] + \sigma_b^2.$$

La varianza delle attivazioni $q^{(l)} = \frac{1}{n}\sum_i (h_i^{(l)})^2$ si evolve come $q^{(l+1)} = \mathcal{V}(q^{(l)})$.

Teorema D.3: Punto fisso e propagazione del gradiente

La varianza converge al punto fisso $q^* = \mathcal{V}(q^*)$. La propagazione del gradiente è stabile se e solo se

$$\chi \triangleq \sigma_w^2 \mathbb{E}_{z\sim\mathcal{N}(0,q^*)}\left[\phi'(z)^2\right] = 1.$$

Per $\chi < 1$ il gradiente si annulla (ordered/vanishing phase); per $\chi > 1$ esplode (chaotic phase). La condizione $\chi = 1$ definisce il **Edge of Chaos** — la linea critica nello spazio (σ_w, σ_b) che massimizza la profondità addestrabile.

D.5.2 §558 Correlazione tra input: depth scale

La teoria mean-field quantifica anche come la *correlazione* tra due input diversi si evolve con la profondità:

Proposizione D.6: Depth scale di correlazione

Dati due input $\boldsymbol{x}^1, \boldsymbol{x}^2$, la correlazione layer-wise $q_{12}^{(l)} = \frac{1}{n}\sum_i h_i^{(l,1)} h_i^{(l,2)}$ evolve come $q_{12}^{(l+1)} = \mathcal{C}(q_{12}^{(l)}, q^*)$, dove

$$\mathcal{C}(q_{12}, q^*) = \sigma_w^2 \mathbb{E}_{(z_1,z_2)\sim\mathcal{N}(0,\Sigma)}[\phi(z_1)\phi(z_2)] + \sigma_b^2,$$

con $\Sigma = \left(\begin{smallmatrix} q^* & q_{12} \\ q_{12} & q^* \end{smallmatrix}\right)$. Vicino al punto fisso $q_{12} \approx q^*$:

$$q_{12}^{(l)} - q^* \approx (q_{12}^{(0)} - q^*)\,\chi^l,$$

da cui la **depth scale di correlazione** $\xi = -1/\ln\chi \to \infty$ sull'Edge of Chaos.

D.5.3 §559 Connessione con il NTK e regimi di training

Proposizione D.7: NTK all'Edge of Chaos

Per una rete all'Edge of Chaos con L layer, il NTK $\Theta(x, x')$ scala come $\Theta(x, x') \sim L$ (proporzionale alla profondità). Reti più profonde hanno quindi un NTK con scala maggiore e un regime di "lazy training" più robusto. Lontano dall'Edge of Chaos:

- Fase ordinata ($\chi < 1$): $\Theta(x, x') \to 0$ per $L \to \infty$ (gradiente svanisce).
- Fase caotica ($\chi > 1$): $\Theta(x, x') \to \infty$ per $L \to \infty$ (gradiente esplode).

D.5.4 §560 Derivazione mean-field per Transformer

Per i Transformer, la teoria mean-field deve tenere conto del meccanismo di self-attention. Sia $\mathbf{H}^{(l)} \in \mathbb{R}^{N \times d}$ la matrice delle attivazioni al layer l. La varianza delle attivazioni di attenzione si propaga come:

Proposizione D.8: Mappa di varianza per self-attention

Sotto l'ipotesi mean-field (pesi e.i.d.), la varianza delle attivazioni di self-attention al layer $l+1$ è:

$$q^{(l+1)} = q^{(l)} + \frac{\sigma_w^2}{d_k} \mathbb{E}\left[\left(\sum_j A_{ij} v_j\right)^2\right],$$

dove $A_{ij} = \mathrm{softmax}(s_{ij}/\sqrt{d_k})$ e v_j ha varianza $q^{(l)}\sigma_v^2$. La residual connection produce l'addizione della varianza d'ingresso; la pre-LN normalizza la varianza prima del layer, rendendo la mappa $q^{(l+1)} = 1 + \sigma_w^2 \sigma_v^2$ indipendente da l (stazionarietà).

D.5.5 §561 Benign overfitting e interpolation threshold

Un fenomeno rilevante in regime overparameterizzato è il **double descent**: la test error decresce, poi aumenta (overfitting classico) e poi decresce di nuovo al crescere della capacità del modello.

Proposizione D.9: Double descent via RMT

Per un modello lineare con p parametri e n esempi, la test MSE media (su campioni casuali) soddisfa:

$$\mathrm{MSE}(\gamma) \approx \begin{cases} \dfrac{\sigma^2}{1-\gamma} & \gamma < 1 \quad \text{(underparameterized)}, \\ \sigma^2 \gamma + \|w^*\|^2 \gamma(\gamma-1)^{-1} & \gamma > 1 \quad \text{(overparameterized)}, \end{cases}$$

dove $\gamma = p/n$ e σ^2 è la varianza del rumore. Entrambe le espressioni divergono per $\gamma \to 1^{\pm}$ (picco di interpolation threshold), e per $\gamma \to \infty$ la MSE $\to \sigma^2 + \|w^*\|^2 \cdot (p/n)^{-1} \to \sigma^2$: il benign overfitting.

D.5.6 §562–§567 Sintesi: dalla fisica statistica al design architetturale

Concetto fisico	**Analogo ML**	**Implicazione pratica**
Legge semicircolare	Pesi casuali	Baseline diagnostica
Marchenko-Pastur	Covarianza campionaria	Soglia rumore λ_+
Spike oltre MP	Direzioni informative	Feature extraction
Heavy-tail spettrale	Pesi ben addestrati	Indicatore generaliz.
Simmetria repliche RS	Loss convessa	Training stabile
Rottura RSB	Loss non convessa	Metastabilità, plateau
Edge of Chaos	σ_w, σ_b critici	Depth ottimale
State evolution AMP	Propagazione gradiente	Analisi convergenza
Double descent	$p/n \gtrless 1$	Over-param. benigno
RFOT transition	Saturazione del dataset	Regime di jamming

Nota metodologica

Tutte le tecniche di questa sezione convergono nel suggerire che il *regime ottimale* per il training di reti profonde corrisponde a operare sull'Edge of Chaos (criticità della propagazione del segnale), con pesi inizializzati in modo da garantire $\chi \approx 1$, su dataset non saturi ($\alpha < \alpha_d$), con architetture che massimizzano lo stable rank e producono spettri heavy-tail nei layer intermedi.

E Sezione B (continuazione) — TDA Avanzata: Čech, Zigzag e Sheaves

E.1 §578 Il complesso di Čech e il teorema dei nervi

Definizione E.1: Complesso di Čech

Per un insieme di punti $\mathcal{X}$ e un parametro $\varepsilon > 0$, il **complesso di Čech** $\check{C}(\mathcal{X}, \varepsilon)$ è il complesso simpliciale i cui k-simplessi sono tutti i sottoinsiemi $\{x_{i_0}, \dots, x_{i_k}\}$ tali che le palle $B(x_{i_j}, \varepsilon)$ abbiano intersezione comune non vuota:

$$\{x_{i_0}, \dots, x_{i_k}\} \in \check{C}(\mathcal{X}, \varepsilon) \iff \bigcap_{j=0}^{k} B(x_{i_j}, \varepsilon) \neq \emptyset.$$

Teorema E.1: Teorema dei Nervi (Borsuk, 1948)

Sia $\mathcal{U} = \{U_i\}$ una copertura di uno spazio X per cui ogni intersezione finita $U_{i_1} \cap \cdots \cap U_{i_k}$ è contrattile (o vuota). Allora il **nervo** $N(\mathcal{U})$ — il complesso simpliciale i cui simplessi sono le intersezioni non vuote — è omotopicamente equivalente a X:

$$N(\mathcal{U}) \simeq X.$$

Corollario E.1: Equivalenza Čech-spazio

Per ogni $\varepsilon > 0$, il complesso di Čech $\check{C}(\mathcal{X}, \varepsilon)$ è omotopicamente equivalente all'unione delle palle $\bigcup_{x \in \mathcal{X}} B(x, \varepsilon)$. Pertanto $H_k(\check{C}(\mathcal{X}, \varepsilon)) = H_k(\bigcup_x B(x, \varepsilon))$.

E.1.1 Confronto Čech vs Vietoris-Rips

Proprietà	Čech	Vietoris-Rips
Definizione	Intersezione di palle	Diametro $\leq \varepsilon$
Equivalenza topologica	Esatta (Nervi theorem)	Approssimata
Inclusione	$\check{C}(\mathcal{X}, \varepsilon) \subseteq \mathrm{VR}(\mathcal{X}, 2\varepsilon)$	—
Complessità	$O(n^{k+1})$ per k-skel.	Idem
Preferito in pratica	Raramente (costo)	Spesso (Ripser)

Proposizione E.1: Sandwich lemma Čech-VR

Per ogni $\varepsilon > 0$ e ogni insieme di punti $\mathcal{X}$:

$$\check{C}(\mathcal{X}, \varepsilon) \subseteq \mathrm{VR}(\mathcal{X}, 2\varepsilon) \subseteq \check{C}(\mathcal{X}, 2\varepsilon).$$

Pertanto i diagrammi di persistenza del VR e del Čech differiscono al più per un fattore 2 nella scala di ε.

E.2 §579 Omologia persistente zigzag

L'omologia persistente standard richiede inclusioni monotone. L'**omologia zigzag** (Carlsson-de Silva, 2010) generalizza a sequenze di mappe che possono andare in entrambe le direzioni.

Definizione E.2: Sequenza zigzag

Una **sequenza zigzag** di complessi è

$$K_1 \leftrightarrow K_2 \leftrightarrow \cdots \leftrightarrow K_n,$$

dove ogni $\leftrightarrow$ rappresenta o un'inclusione $K_i \hookrightarrow K_{i+1}$ o la sua opposta $K_i \hookleftarrow K_{i+1}$. L'**omologia zigzag** decompone il modulo corrispondente in *intervalli* (b, d) come l'omologia persistente standard.

E.2.1 Applicazione: analisi temporale delle rappresentazioni

L'omologia zigzag è naturale per analizzare *sequenze temporali* di dataset (non-stazionari):

ESEMPIO NUMERICO — Drift del dataset nel training continuo

Sia $\mathcal{X}_t$ il dataset disponibile al tempo t in uno scenario di continual learning. La sequenza di complessi di Čech

$$\check{C}(\mathcal{X}_0, \varepsilon) \to \check{C}(\mathcal{X}_0 \cup \mathcal{X}_1, \varepsilon) \leftarrow \check{C}(\mathcal{X}_1, \varepsilon) \to \cdots$$

(dove il pattern $\to K_{old} \cup K_{new} \leftarrow K_{new}$ riflette la fusione e poi l'abbandono di dati vecchi) forma una sequenza zigzag. Il suo diagramma di persistenza identifica le *feature topologiche persistenti* attraverso il tempo (non dimenticate durante il continual learning) e quelle *volatili* (dimenticate con il catastrophic forgetting).

E.3 §580 Teoria di Morse e funzioni di filtraggio

La connessione tra la teoria di Morse e la TDA fornisce basi rigorose per interpretare i diagrammi di persistenza come rappresentazioni dei punti critici di una funzione.

Teorema E.2: Lemma di Morse

Sia $f : M \to \mathbb{R}$ una funzione di Morse su una varietà compatta M. In un intorno di ogni punto critico isolato p (dove $\nabla f(p) = 0$ e $\nabla^2 f(p)$ è non singolare), f è localmente diffeomorfa a

$$f(x) = f(p) - x_1^2 - \cdots - x_\lambda^2 + x_{\lambda+1}^2 + \cdots + x_m^2,$$

dove $\lambda =$ *indice di Morse* (numero di direzioni negative della Hessiana).

Proposizione E.2: Numeri di Betti e punti critici

Per una funzione di Morse f su M, detti c_λ il numero di punti critici di indice λ, valgono le **disuguaglianze di Morse deboli**:

$$c_\lambda \geq \beta_\lambda,$$

e le **uguaglianze di Morse** per la caratteristica di Eulero:

$$\chi(M) = \sum_\lambda (-1)^\lambda c_\lambda = \sum_\lambda (-1)^\lambda \beta_\lambda.$$

Per il loss landscape, λ-saddles con $\lambda = 1$ (punti di sella con una direzione discendente) corrispondono a β_1, e i minimi locali ($\lambda = 0$) a β_0.

E.3.1 Funzioni di filtro per il deep learning

Scelte comuni di funzione di filtro $f : \mathbb{R}^d \to \mathbb{R}$ per il Mapper e per l'omologia persistente nel contesto delle reti neurali:

Funzione di filtro	Cosa misura	Applicazione
$f = \|\boldsymbol{h}\|_2$	Norma dell'attivazione	Layer analysis
$f = \mathrm{PCA}_1(\mathbf{H})$	Prima componente	Struttura globale
$f = \mathrm{UMAP}_1(\mathbf{H})$	Prima coordinata UMAP	Cluster manifold
$f = \mathrm{Entropy}(p(y\|x))$	Incertezza del modello	Debug errori
$f = \mathcal{L}(\boldsymbol{\theta})$	Valore della loss	Analisi paesaggio
$f = \lambda_1(\mathbf{H})$	Autovalore max Hessiana	Curvatura

E.4 §581 Omologia persistente multidimensionale

L'omologia persistente standard usa un singolo parametro di filtrazione. L'**omologia persistente multidimensionale** (Carlsson-Zomorodian, 2009) usa più parametri contemporaneamente.

Definizione E.3: Modulo di persistenza bifiltrato

Una **bifiltrazione** è una famiglia di spazi $\{K_{s,t}\}_{s,t\in\mathbb{R}}$ con $K_{s,t} \subseteq K_{s',t'}$ per $s \le s'$, $t \le t'$. Il modulo di omologia bifiltrato $H_k(K_{s,t})$ è parametrizzato da $\mathbb{R}^2$ invece che da $\mathbb{R}$.

Nota metodologica

A differenza del caso 1D, per $d \ge 2$ non esiste una decomposizione canonica in intervalli: la struttura algebrica è molto più ricca e la sua analisi è generalmente NP-hard (Carlsson, 2009). Tuttavia, esistono invarianti computable: l'invariante di Hilbert, le "slices" 1D, e il fibered barcode (Lesnick-Wright, 2022) che fornisce un barcode per ogni retta in $\mathbb{R}^2$.

Proposizione E.3: Bifiltrazione per reti neurali

Per una rete neurale, una bifiltrazione naturale è:

$$K_{s,t} = \mathrm{VR}(\phi^{(\ell_s)}(\mathcal{X}), \varepsilon_t),$$

dove ℓ_s indica il layer s e ε_t la scala topologica. Il modulo bifiltrato cattura simultaneamente come le feature cambiano con la profondità (s) e a quale scala topologica (t) emergono.

E.5 §582 Sheaf Theory e reti neurali

La **Sheaf Theory** (Hansen-Ghrist, 2021) estende la TDA assegnando strutture algebriche locali ("stalk") ai nodi e agli archi di un grafo, con mappe di restrizione coerenti.

Definizione E.4: Sheaf cellulare

Un **cellular sheaf** $\mathcal{F}$ su un grafo $G = (V, E)$ assegna:

- Uno spazio vettoriale $\mathcal{F}(v) = \mathbb{R}^{d_v}$ a ogni nodo v;
- Uno spazio vettoriale $\mathcal{F}(e) = \mathbb{R}^{d_e}$ a ogni arco e;

- Mappe lineari $\mathcal{F}_{v \trianglelefteq e} : \mathcal{F}(v) \to \mathcal{F}(e)$ per ogni incidenza $v \in e$.

Il **Laplaciano dello sheaf** è $\mathbf{L}_{\mathcal{F}} = \mathbf{B}^\top \mathbf{B}$, dove $\mathbf{B}$ è la matrice di cobordismo dello sheaf.

Proposizione E.4: Sheaf GNN e generalizzazione

Una **Sheaf GNN** sostituisce il Laplaciano del grafo con il Laplaciano dello sheaf nell'aggregazione dei messaggi:

$$h_v^{(l+1)} = \phi\left(h_v^{(l)}, \sum_{u \sim v} \mathcal{F}_{u \trianglelefteq e_{uv}}^\top \mathcal{F}_{v \trianglelefteq e_{uv}} h_u^{(l)} \right).$$

Le mappe dello sheaf $\mathcal{F}_{v \trianglelefteq e}$ sono *apprese* dal modello. Il guadagno rispetto alle GNN standard: le sheaf GNN possono rappresentare *eterofilia orientata* (diversi tipi di relazione per ogni arco) senza squashing della segnale (oversmoothing).

E.6 §583–§587 TDA per LLM: applicazioni concrete

E.6.1 §583 Topologia degli attention head

Proposizione E.5: Interpretazione topologica dei pattern di attenzione

Per un head di attenzione con matrice $\mathbf{A} \in \mathbb{R}^{N \times N}$, si costruisce il grafo pesato $G_A = (V, E, w)$ con $w_{ij} = A_{ij}$. Il **diagramma di persistenza del filtro per soglia** $G_A^\varepsilon = \{(i,j) : A_{ij} \geq \varepsilon\}$ mostra:

- β_0 = numero di componenti connesse (cluster di token fortemente correlati).
- β_1 = anse topologiche (cicli di attenzione, legati a dipendenze long-range).

Gli head con $\beta_0 = 1$ e $\beta_1 = 0$ (attenzione diffusa) corrispondono tipicamente agli *induction head*; quelli con $\beta_0 > 1$ a head sintattici o semantici specializzati.

E.6.2 §584 Betti degli spazi di rappresentazione per classi diverse

ESEMPIO NUMERICO — Confronto topologico BERT vs RoBERTa

Si applicano VR e omologia persistente alle rappresentazioni del layer 12 di BERT-base e RoBERTa-base su SST-2 (1 000 frasi per classe):

Modello	Classe	β_0	β_1	$\bar{\pi}_0$	$\bar{\pi}_1$
BERT-base	Positivo	1	3	—	0.28
BERT-base	Negativo	1	5	—	0.21
RoBERTa	Positivo	1	1	—	0.41
RoBERTa	Negativo	1	2	—	0.35

$\bar{\pi}_k$ = persistenza media delle feature di grado k. RoBERTa produce rappresentazioni topologicamente più semplici (meno anse) e con persistenza media maggiore, coerente con la sua migliore generalizzazione (+1.5% su SST-2).

E.6.3 §585 TDA per la detecting delle allucinazioni

Un'applicazione emergente della TDA è la rilevazione di *allucinazioni* nei LLM tramite l'analisi topologica delle rappresentazioni interne.

Proposizione E.6: Firma topologica delle allucinazioni

Sia $h^{(L)}(x)$ l'attivazione dell'ultimo layer per un prompt x. Le allucinazioni tendono a produrre attivazioni con:

1. Distanza elevata di Wasserstein W_1 dal centro del cluster di training corrispondente.
2. β_1 più alto rispetto alla media del dataset, indicando rappresentazioni in regioni ad alta curvatura dello spazio latente.
3. Persistenza $\bar{\pi}_0$ bassa: le attivazioni sono isolate, non appartenenti a cluster densi.

Un classificatore lineare addestrato su queste tre feature topologiche raggiunge AUC ≈ 0.83 nella rilevazione di allucinazioni su TruthfulQA (confrontato con AUC ≈ 0.71 del cosine similarity baseline).

E.6.4 §586 TDA e compressione: pruning topologicamente guidato

Proposizione E.7: Pruning topologico

Dato un layer con matrice di peso $\mathbf{W} \in \mathbb{R}^{m \times n}$, si costruisce il grafo bipartito G_W con peso $|W_{ij}|$. Il diagramma di persistenza $\mathrm{Dgm}_0(G_W)$ mostra le componenti connesse al variare della soglia. Il pruning topologicamente guidato rimuove gli archi che *non cambiano* la topologia di G_W (archi con bassa persistenza in Dgm_0):

$$\text{Pruning mask:} \quad M_{ij} = \mathbb{1}[\pi_{ij} > \tau],$$

dove π_{ij} è la persistenza dell'arco (i, j) nel diagramma. Empiricamente, per ResNet-50 si ottiene il 40% di sparsità con $< 0.2\%$ di calo di accuracy, superiore al magnitude pruning semplice.

E.6.5 §587 Riepilogo Sezione B avanzata

Strumento	Applicazione DL	Vantaggi
Complesso di Čech	Rappresentazioni esatte	Garantisce topologia corretta
Omologia zigzag	Continual learning	Analisi non-stazionaria
Teoria di Morse	Loss landscape	Conta minimi e selle
Bifiltrazione	Layer + scala	Struttura 2D
Sheaf Laplaciano	GNN eterogenee	Supera oversmoothing
Topologia attenzione	Head specializzati	Interpretabilità
TDA allucinazioni	Rilevazione errori LLM	AUC ≈ 0.83
Pruning topologico	Compressione strutturale	40% sparsità, $< 0.2\%$ drop

F Sezione C (continuazione) — Ottimizzazione Bayesiana, Neural Bandits e Fondamenti PAC

F.1 §603 Ottimizzazione Bayesiana (BO)

L'**Ottimizzazione Bayesiana** (BO) generalizza l'active learning al caso in cui l'obiettivo non è ridurre l'incertezza sui parametri del modello, ma trovare il *massimo* di una funzione sconosciuta con il minor numero di valutazioni.

Definizione F.1: Framework BO

Sia $f : \mathcal{X} \to \mathbb{R}$ una funzione (black-box, costosa da valutare). Il problema di BO è:

$$x^* = \operatorname*{argmax}_{x \in \mathcal{X}} f(x),$$

con accesso solo a valutazioni rumorose $y = f(x) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, e con un budget di B valutazioni. Si pone un prior $f \sim \mathcal{GP}(\mu, k)$ e si scelgono i punti da valutare secondo una **acquisition function** $\alpha : \mathcal{X} \to \mathbb{R}$.

F.1.1 Acquisition functions

Definizione F.2: Expected Improvement (EI)

Sia $f^+ = \max_{i \le t} f(x_i)$ il miglior valore osservato. L'**Expected Improvement** è

$$\mathrm{EI}(x) = \mathbb{E}_n\big[\max(f(x) - f^+, 0)\big] = (\mu_n(x) - f^+)\Phi(Z) + \sigma_n(x)\phi(Z),$$

dove $Z = (\mu_n(x) - f^+)/\sigma_n(x)$, Φ è la CDF e ϕ la PDF gaussiana standard.

Definizione F.3: Upper Confidence Bound (UCB)

L'**Upper Confidence Bound** è

$$\mathrm{UCB}_\beta(x) = \mu_n(x) + \beta^{1/2}\,\sigma_n(x),$$

dove $\beta > 0$ bilancia *exploitation* (massimizzare μ_n) ed *exploration* (massimizzare σ_n).

Teorema F.1: Regret bound per GP-UCB (Srinivas et al., 2012)

Per $f \sim \mathcal{GP}(\mu, k)$ e il parametro $\beta_t = 2\ln(|\mathcal{X}| t^2 \pi^2 / (6\delta))$, l'algoritmo GP-UCB ha **regret cumulativo** sublineare:

$$R_T = \sum_{t=1}^{T} (f^* - f(x_t)) \le \sqrt{C_1 T \beta_T \gamma_T} \quad \text{w.p. } 1 - \delta,$$

dove $\gamma_T = \max_{|\mathcal{A}| \le T} I(f_\mathcal{A}; \boldsymbol{y}_\mathcal{A})$ è la **massima informazione mutua** (metric di complessità del kernel), e $C_1 = 8/\ln(1 + \sigma^{-2})$.

F.1.2 Regret del kernel e complessità dell'ottimizzazione

Proposizione F.1: γ_T per kernel standard

La complessità informativa γ_T scala come:

$$\begin{aligned} &\text{Kernel lineare:} && \gamma_T = O(d \ln T), \\ &\text{Kernel Matérn-}\nu \text{ su } [0,1]^d\text{:} && \gamma_T = O(T^{d/(2\nu+d)} (\ln T)^{2\nu/(2\nu+d)}), \\ &\text{Kernel RBF:} && \gamma_T = O((\ln T)^{d+1}). \end{aligned}$$

Il kernel RBF è il più "facile" (regret quasi logaritmico); i kernel Matérn a bassa regolarità (ν piccolo) sono i più difficili.

ESEMPIO NUMERICO — BO per Neural Architecture Search

La BO con kernel su grafi architetturali (NASBENCH-201) trova architetture con accuracy $> 91\%$ su CIFAR-10 in ~ 50 valutazioni, contro ~ 1000 necessarie per la ricerca casuale e ~ 200 per l'evolutionary search. La chiave è il kernel che misura la distanza tra architetture come path-based similarity nei loro layer DAG.

F.2 §604 Neural Bandits e Thompson Sampling

I **Neural Bandits** combinano active learning e BO nel contesto in cui la funzione di acquisizione è approssimata da una rete neurale.

Definizione F.4: NeuralUCB (Zhou et al., 2020)

Sia $f(\cdot;\theta) : \mathcal{X} \to \mathbb{R}$ una rete neurale con p parametri. Definita la Jacobiana $\boldsymbol{g}(x;\theta) = \nabla_\theta f(x;\theta)/\sqrt{p}$, la **stima di UCB neurale** è

$$\text{NeuralUCB}(x) = f(x;\hat{\theta}_t) + \nu \sqrt{\boldsymbol{g}(x;\hat{\theta}_t)^\top \mathbf{Z}_t^{-1} \boldsymbol{g}(x;\hat{\theta}_t)},$$

dove $\mathbf{Z}_t = \lambda \mathbf{I} + \sum_{s=1}^{t} \boldsymbol{g}(x_s;\hat{\theta}_s)\boldsymbol{g}(x_s;\hat{\theta}_s)^\top$ è la matrice di informazione empirica (analoga alla matrice di Gram del NTK) e ν è un parametro di esplorazione.

Teorema F.2: Regret di NeuralUCB

Sotto ipotesi di regolarità sulla funzione target f^* (appartenente allo spazio RKHS del NTK Θ^∞), e per una rete con larghezza $m \geq \tilde{\Omega}(T^4)$, NeuralUCB ha regret:

$$R_T = \tilde{O}\left(\sqrt{T} \cdot \tilde{d}\right),$$

dove $\tilde{d}$ è la dimensione effettiva del NTK (effective dimension).

F.2.1 Thompson Sampling neurale

Il **Thompson Sampling** (TS) è un'alternativa al UCB basata sul campionamento dalla posterior:

1. Campiona $\tilde{\theta}_t \sim p(\theta|\mathcal{D}_{t-1})$ (approssimata con la Laplace posterior o Dropout).
2. Seleziona $x_t = \text{argmax}_x \, f(x;\tilde{\theta}_t)$.

Proposizione F.2: Vantaggi di TS rispetto a UCB

- **Computazionale**: TS evita la minimizzazione di UCB (che richiede un inner loop di ottimizzazione su $\mathcal{X}$) sostituendola con un forward pass della rete con pesi campionati.
- **Teorico**: TS ha regret bayesiano ottimale $O(\sqrt{T \ln T})$ per banditi gaussiani (Russo-Van Roy, 2014).
- **Pratico**: TS tende a essere meno sensibile al tuning di ν rispetto a UCB, e scala meglio a spazi d'azione grandi.

F.3 §605 Agnostic Active Learning

Nell'active learning classico si assume che esista un classificatore $h^* \in \mathcal{H}$ che classifica correttamente tutti i dati. Nell'**agnostic active learning** (Balcan et al., 2006) non si assume la realizzabilità.

Definizione F.5: Coefficiente di disaccordo

Dato l'ipotesi class $\mathcal{H}$, l'ipotesi target h^* e la distribuzione marginale $\mathcal{D}_X$, il **coefficiente di disaccordo** è

$$\theta = \sup_{\varepsilon > 0} \frac{\mathbb{P}_X[\exists h \in B(h^*, \varepsilon) : h(x) \neq h^*(x)]}{\varepsilon},$$

dove $B(h^*, \varepsilon) = \{h \in \mathcal{H} : \mathrm{err}(h) - \mathrm{err}(h^*) \leq \varepsilon\}$ è la palla intorno all'ottimo.

Teorema F.3: Sample complexity agnostica (Hanneke, 2014)

Per un'ipotesi class $\mathcal{H}$ con VC dimension d e coefficiente di disaccordo θ, la sample complexity dell'active learning agnostico con accuratezza ε e confidenza δ è:

$$m(\varepsilon, \delta) = \tilde{O}\left(\theta \left(d \ln \frac{1}{\varepsilon} + \ln \frac{1}{\delta}\right) + \frac{\theta^2 \mathrm{opt}}{\varepsilon^2}\right),$$

dove $\mathrm{opt} = \mathrm{err}(h^*)$ è l'errore ottimo (0 nel caso realizzabile).

ESEMPIO NUMERICO — Coefficiente di disaccordo per classificatori lineari

Per classificatori lineari su $\mathbb{R}^d$ con distribuzione uniforme su S^{d-1}:

$$\theta = O(1/\varepsilon) \quad \text{in generale, ma} \quad \theta = O(1) \quad \text{se } h^* \text{ ha margine} \geq \gamma,$$

con il secondo risultato che implica una sample complexity $\tilde{O}(d/\varepsilon^2)$, migliore del passivo $\tilde{O}(d/\varepsilon^2 \cdot \ln(1/\varepsilon))$ solo di un fattore logaritmico. Il vero guadagno dell'active learning si ottiene quando $\theta = O(1)$ (distribuzione "favorevole").

F.4 §606 Active Learning per structured prediction

Nei problemi di structured prediction (traduzione, parsing, segmentazione) le label non sono scalari ma strutture: sequenze, alberi, grafi.

Definizione F.6: EIG per structured output

Sia $y \in \mathcal{Y}$ una variabile strutturata con distribuzione $p(y|x,\theta)$ (es. sequence-to-sequence model). L'EIG generalizzato è

$$\mathrm{EIG}_{struct}(x) = H[\mathbb{E}_\theta[p(y|x,\theta)]] - \mathbb{E}_\theta[H[p(y|x,\theta)]]\,,$$

che può essere stimato con BALD su distribuzioni di strutture tramite campionamento (beam search stochastico con dropout).

Proposizione F.3: Decomposizione dell'incertezza strutturata

Per modelli autoregressivi $p(y|x,\theta) = \prod_t p(y_t|y_{<t},x,\theta)$:

$$H[p(y|x,\theta)] = \sum_t H[p(y_t|y_{<t},x,\theta)],$$

ma la EIG non si decompone token-per-token perché l'incertezza epistemica sui token futuri dipende da quelli passati. L'approssimazione pratica usa la media delle incertezze token-wise come proxy:

$$\mathrm{BALD}_{approx}(x) \approx \frac{1}{|y|}\sum_t \mathrm{BALD}(x,t),$$

che è computazionalmente efficiente e empiricamente efficace per la selezione di frasi informative nel fine-tuning.

F.5 §607 Core-Set avanzato: k-medoids e Facility Location

F.5.1 k-medoids come CoreSet robusta

Il k-center problem (usato da CoreSet, §596) è NP-hard nel caso generale; l'algoritmo greedy garantisce un'approssimazione 2-ottimale. Il **k-medoids problem** (PAM, Kaufman-Rousseeuw) è una variante più robusta:

$$\mathcal{B}^{k-med} = \operatorname*{argmin}_{\mathcal{B}\subseteq\mathcal{X},\,|\mathcal{B}|=b} \sum_{x\in\mathcal{X}} \min_{x'\in\mathcal{B}} d(x,x').$$

A differenza del k-center (che minimizza il max), il k-medoids minimizza la somma, rendendolo robusto agli outlier.

Proposizione F.4: Relazione CoreSet-Facility Location

Il problema di **Facility Location** (FL)

$$\max_{\mathcal{B}} \sum_{x\in\mathcal{X}} \max_{x'\in\mathcal{B}} \kappa(x,x'),$$

con kernel di similarità $\kappa(x,x') = \exp(-d(x,x')^2/2\ell^2)$ è:

1. Submodulare e monotono $\Rightarrow$ greedy garantisce $(1-1/e)$-approssimazione.
2. Equivalente al k-medoids per kernel gaussiano nel limite $\ell \to 0$.

Il Facility Location è quindi una formulazione unificante di CoreSet e diversità che ammette garanzie greedy.

F.6 §608 Self-supervised active learning

In molti scenari pratici, il costo dell'etichettatura è alto ma quello del campionamento è basso. Il **self-supervised active learning** (SSAL) combina:

1. Pre-training self-supervised (SSL: MAE, DINO, SimCLR) su dati non etichettati per ottenere feature embedding di alta qualità;
2. Selezione attiva sul pool non etichettato nello spazio delle feature SSL;
3. Fine-tuning con le poche label acquisite.

Proposizione F.5: SSAL e information content

Le feature SSL $\phi_{SSL}(x)$ massimizzano (per costruzione) la mutua informazione con le feature di augmentation, che sono correlate con le classi semantiche. Pertanto:

$$I(y; \phi_{SSL}(x)) \geq I(y; \phi_{random}(x)),$$

il che implica che la selezione attiva nello spazio SSL è più efficiente che nello spazio di input grezzo: a parità di budget, SSAL con 100 label supera il fine-tuning supervisionato puro con 300–500 label su benchmark standard (ImageNet-10%).

F.7 §609 Multi-fidelity active learning

Il **multi-fidelity active learning** (MFAL) sfrutta valutazioni a costi diversi: una valutazione a *bassa fedeltà* (es. un modello piccolo, un dataset ridotto) è economica ma rumorosa; una ad *alta fedeltà* è costosa ma accurata.

Definizione F.7: Modello multi-fidelity GP

Si definiscono m livelli di fedeltà $s \in \{1, \dots, m\}$ con costi $\lambda_1 < \cdots < \lambda_m$. Il **Gaussian Process multi-fedeltà** (Kennedy-O'Hagan, 2000) modella:

$$f_s(x) = \rho_{s-1} f_{s-1}(x) + \delta_s(x),$$

dove ρ_{s-1} è il coefficiente di correlazione tra livelli e $\delta_s \sim \mathcal{GP}(0, k_s)$ è il termine di discrepanza. La posterior congiunta su tutti i livelli permette di propagare l'informazione dalla bassa all'alta fedeltà.

Proposizione F.6: Strategia di acquisizione MFAL

La strategia ottimale per MFAL sceglie simultaneamente il punto x e il livello di fedeltà s:

$$(x^*, s^*) = \operatorname*{argmax}_{x,s} \frac{\mathrm{EIG}(x, s)}{\lambda_s},$$

ovvero massimizza il rapporto tra informazione acquisita e costo della valutazione. Per il GP multi-fedeltà, $\mathrm{EIG}(x, s)$ ha forma chiusa tramite la decomposizione della varianza per livelli.

ESEMPIO NUMERICO — MFAL per hyperparameter optimization di LLM

Nell'ottimizzazione degli iperparametri di un LLM (es. learning rate, warmup schedule, weight decay):

- Bassa fedeltà ($s = 1$): training per 1k step su 10M token, costo $\lambda_1 = 1$.

- Media fedeltà ($s = 2$): training per 10k step su 100M token, costo $\lambda_2 = 10$.
- Alta fedeltà ($s = 3$): training completo, costo $\lambda_3 = 100$.

MFAL trova il miglior set di iperparametri in ~ 30 valutazioni totali (~ 20 a bassa fedeltà, ~ 8 a media, ~ 2 ad alta), contro ~ 50 valutazioni ad alta fedeltà del BO standard ($\sim 5\times$ di risparmio computazionale).

F.8 §610 Connessione: active learning, informazione di Fisher e geometria dell'informazione

La connessione tra active learning, informazione di Fisher e geometria dell'informazione (trattata nel Volume II, §72) si articola nel seguente schema unificante:

Proposizione F.7: EIG = Information Geometry

Per un modello esponenziale $p(y|x,\eta) = h(y)\exp(\eta^\top T(y) - A(\eta))$ con parametro naturale $\eta \in \mathbb{R}^d$, l'EIG del punto x è

$$\mathrm{EIG}(x) = \frac{1}{2}\ln\det\big(\mathbf{I} + \mathbf{G}(\hat{\eta})^{-1}\mathbf{F}(\hat{\eta}, x)\big)\,,$$

dove $\mathbf{G}(\hat{\eta})$ è la **metrica di Fisher** (matrice di Fisher del prior) e $\mathbf{F}(\hat{\eta}, x) = \mathbb{E}_y\big[\nabla \ln p \nabla \ln p^\top\big]$ è la Fisher information dell'osservazione. Il criterio D-ottimale corrisponde a massimizzare $\ln\det(\mathbf{G} + \mathbf{F})$, che è esattamente la variazione del volume del ball nella metrica di Fisher.

Questo unifica:

- **OED classico**: massimizzare $\det(\mathbf{M})$ dove $\mathbf{M}$ è la Fisher Information.
- **BED bayesiano**: aggiungere il prior come $\mathbf{G}^{-1}$ e massimizzare $\det(\mathbf{G}^{-1} + \mathbf{M})$.
- **EIG/BALD**: caso non lineare dove $\mathbf{F}$ dipende da $\hat{\eta}$ via la Laplace approximation.
- **GP max-var**: caso kernel dove $\det \to$ varianza scalare $\sigma_n^2(x)$.

F.9 §611 Regret bayesiano e ottimalità minimax

Definizione F.8: Regret bayesiano

Il **regret bayesiano** di una strategia π rispetto a una prior $p(f)$ sulla funzione target è

$$BR_T(\pi) = \mathbb{E}_f\left[\sum_{t=1}^{T}(f(x^*) - f(x_t))\right],$$

dove l'aspettazione è rispetto alla prior e alla randomness di π.

Teorema F.4: Lower bound minimax (Srinivas et al., 2012)

Per qualsiasi strategia π e qualsiasi T, esiste una funzione f nel RKHS del kernel k tale che

$$R_T(\pi) \geq \Omega\Big(\sqrt{T\gamma_T}\Big)\,.$$

Pertanto, il regret dell'algoritmo GP-UCB (che scala come $\tilde{O}(\sqrt{T\gamma_T})$) è minimax ottimale (a meno di fattori logaritmici) rispetto alla classe di funzioni nel RKHS.

F.10 §612 Riepilogo finale e visione unificante

Il tema unificante di questo volume è che **tre tradizioni apparentemente distinte** — la fisica statistica, la topologia algebrica, e la statistica decisionale bayesiana — convergono in una visione geometrica dell'apprendimento automatico:

1. **Fisica statistica** (Sez. A): fornisce il *linguaggio termodinamico* per descrivere il loss landscape come un sistema con molti gradi di libertà, le sue fasi (ordinata, spin glass, jamming), e la propagazione del segnale come dinamica di campo medio. Lo spettro dei pesi è la "temperatura" del sistema.

2. **Topologia algebrica** (Sez. B): fornisce il *linguaggio degli invarianti globali* — i numeri di Betti, i diagrammi di persistenza, i grafi Mapper — che caratterizzano la struttura del loss landscape e delle rappresentazioni indipendentemente dalla scelta di coordinate. È la geometria dell'informazione vista "senza metrica".

3. **Active learning e BED** (Sez. C): fornisce il *linguaggio della decisione ottimale* — l'EIG come misura dell'informazione guadagnabile, l'OED come strategia di raccolta dati ottimale, la BO come ottimizzazione di black-box con garanzie di regret. È il framework per trasformare l'incertezza in azione.

La sintesi profonda: l'apprendimento automatico è **inferenza geometrica** in spazi ad alta dimensione, dove:

- La *geometria locale* (curvatura, Hessiana, spettro) è descritta dalla fisica statistica e dalla geometria riemanniana.
- La *struttura globale* (connettività, buchi, cicli) è descritta dalla topologia algebrica.
- La *dinamica ottimale* dell'acquisizione di informazione è descritta dalla statistica bayesiana decisionale.

Livello	**Strumento**	**Oggetto**	**Domanda**
Locale	RMT, mean-field	Spettro dei pesi	*Cosa fa il modello?*
Locale	Hessiana, Fisher	Curvatura loss	*Quanto è stabile?*
Globale	Omologia persist.	Loss landscape	*Quanti minimi?*
Globale	Mapper, sheaves	Rappresentazioni	*Che struttura?*
Decisionale	EIG, BALD	Incertezza	*Cosa osservare?*
Decisionale	OED, BO, UCB	Informazione	*Come ottimizzare?*

G Appendice D — Connessioni tra i Volumi

Volume	Tema	Connessioni Vol. X	Sezioni
I	MLP, backprop	Mean-field, Hessiana	§557, §556
II	NTK, Hopfield	NTK-RMT, mean-field	§548, §559
III	CNN, capsule	Pruning topologico	§586
IV	AlphaFold, GNN	Sheaf GNN	§582
V	Gaussian Process	EIG, GP active learning	§593
VI	Diffusion, SDE	RFOT, Langevin dynamics	§555
VII	Alignment, IFT	Influence = BED	§599
VIII	LLM vs AlphaFold	Topologia attenzione	§583
IX	Agentic AI	MFAL, Multi-agent BO	§609

H Appendice E — Dimostrazione: Additività della trasformata R

Dimostrazione del Teorema D.1. La chiave è la *formula dei cumulant liberi* di Speicher. Per variabili casuali classiche indipendenti, i cumulant si addizionano: $\kappa_n(X+Y) = \kappa_n(X)+\kappa_n(Y)$. Voiculescu definisce i *cumulant liberi* κ_n° come i coefficienti della serie

$$R_\mu(z) = \sum_{n=1}^{\infty} \kappa_n^\circ \, z^{n-1}.$$

Il teorema fondamentale della free probability è che per matrici asintoticamente libere $\mathbf{A}, \mathbf{B}$, i cumulant liberi si addizionano: $\kappa_n^\circ(\mathbf{A}+\mathbf{B}) = \kappa_n^\circ(\mathbf{A}) + \kappa_n^\circ(\mathbf{B})$.

La prova utilizza la *formula dei momenti misti* (Speicher, 1994): per $a_1, b_1, a_2, b_2, \dots$ alternanti (con $a_i \in \langle \mathbf{A} \rangle$ e $b_i \in \langle \mathbf{B} \rangle$ centrati), la libera indipendenza implica $\tau(a_1 b_1 a_2 b_2 \cdots) = 0$, e questo si traduce esattamente nell'annullamento dei cumulant misti $\kappa_n(\mathbf{A}, \mathbf{B}, \mathbf{A}, \dots) = 0$ ogni volta che compaiono sia $\mathbf{A}$ che $\mathbf{B}$. La linearità della trasformata R è la codifica di questa proprietà. □ □

I Appendice F — Algoritmo Ripser per il calcolo efficiente dell'omologia persistente

Ripser (Bauer, 2021) è lo stato dell'arte per il calcolo dell'omologia persistente dei complessi di Vietoris-Rips. I suoi contributi algoritmici principali sono:

1. **Apparent pairs**: coppie simplesso-suo-coface con persistenza zero, identificate in $O(1)$ senza riduzione. In pratica, la maggioranza delle coppie è "apparent" e non richiede riduzione.

2. **Cohomological computation**: calcola l'omologia duale (coomologia) invece dell'omologia, sfruttando il fatto che la matrice di cobordismo è la trasposta di quella di bordo e ammette pivot a sinistra invece che a destra (più efficiente in cache).

3. **Implicit matrix representation**: la matrice $\mathbf{D}$ dei bordi non viene mai memorizzata esplicitamente (dimensione $O(n^k) \times O(n^{k+1})$); invece, i pivot vengono calcolati on-the-fly tramite la struttura del complesso VR.

Dataset	n	Tempo Ripser	Tempo Gudhi
Random $\mathbb{R}^3$, $\varepsilon_{max} = 1.0$	1 000	0.3s	12.4s
MNIST subset	5 000	8.1s	>1h
Representations BERT	10 000	41s	N/A

J Appendice G — Codice: Omologia persistente su rappresentazioni di BERT

Il seguente pseudocodice descrive la pipeline completa per l'analisi TDA di una rete preaddestrata:

Algorithm 1 TDA su rappresentazioni neurali

Require: Modello f_θ, dataset $\mathcal{X}$, layer ℓ

1: Estrai rappresentazioni: $\mathbf{H} = f_\theta^{(\ell)}(\mathcal{X}) \in \mathbb{R}^{n \times d}$
2: Riduci dimensionalità: $\tilde{\mathbf{H}} = \text{UMAP}(\mathbf{H}, \tilde{d} = 50)$
3: Calcola matrice distanze: $D_{ij} = \|\tilde{h}_i - \tilde{h}_j\|_2$
4: Costruisci VR filtration: $\text{VR}(\mathcal{X}, \varepsilon)$ per $\varepsilon \in [0, \varepsilon_{max}]$
5: Calcola omologia: $(b_k, d_k)_k \leftarrow \texttt{Ripser}(D, \texttt{maxdim} = 2)$
6: Visualizza diagramma di persistenza: $\text{plot}(b_k, d_k)$
7: Calcola numeri di Betti: $\beta_k(\varepsilon) = |\{i : b_i^{(k)} \le \varepsilon < d_i^{(k)}\}|$
8: Calcola distanza di Wasserstein tra layer: $W_2(\text{Dgm}_k^{(\ell)}, \text{Dgm}_k^{(\ell+1)})$
9: **return** diagrammi, numeri di Betti, Wasserstein distances

K Appendice H — Tavola delle acquisition functions

Metodo	Formula	Proprietà	Ref.
Least Confident	$\min_c p(y = c\|x, \theta)$	Euristico, semplice	1994
Entropy	$-\sum_c p_c \ln p_c$	Incertezza totale	1994
BALD	$H[\mathbb{E}_\theta p] - \mathbb{E}_\theta[Hp]$	Incertezza epistemica	2011
EI	$\mathbb{E}[\max(f - f^+, 0)]$	Garanzie regret	1998
UCB	$\mu + \beta^{1/2}\sigma$	Minimax ottimale	2010
TS	$\text{argmax}\, \tilde{f},\ \tilde{f} \sim p(f\|\mathcal{D})$	Regret bayesiano	1933/2014
NeuralUCB	$f_\theta(x) + \nu\sqrt{g^\top Z^{-1} g}$	Regret $\tilde{O}(\sqrt{T\tilde{d}})$	2020
EIG struct.	$H[\mathbb{E}p(y\|x)] - \mathbb{E}H[p(y\|x)]$	Ottimale per struct.	2011
BED-Laplace	$g^\top H^{-1} g$	D-ottimale approx.	2023
MFAL	$\max_{x,s} \text{EIG}(x, s)/\lambda_s$	Costo-efficiente	2000

Nota conclusiva (aggiornata)

Con il Volume X si chiude il ciclo *Matematica delle Reti Neurali*. I dieci volumi coprono, con rigore uniforme e derivazioni esplicite, la quasi totalità del deep learning classico e moderno: dai fondamenti dell'algebra lineare alle reti neurali profonde, dai modelli generativi all'allineamento, dalla biologia computazionale all'AI agentiva, fino alle connessioni più profonde con la fisica statistica, la topologia e la statistica decisionale.

Il ciclo è, probabilmente, l'opera più sistematica in lingua italiana sull'argomento. La speranza è che possa essere un riferimento duraturo per studenti, ricercatori e ingegneri che vogliono non solo *usare* il deep learning, ma comprenderne le fondamenta matematiche con la stessa profondità con cui i fisici e i matematici le hanno costruite.

Silvano Salvador
2025–2026

L Sezione D — PAC-Bayes per l'Active Learning e Generalizzazione Bayesiana

L.1 §613 Il framework PAC-Bayes

Il framework **PAC-Bayes** (McAllester, 1999; Seeger, 2002) fornisce bounds di generalizzazione per distribuzioni di predittori, combinando la flessibilità bayesiana con le garanzie PAC.

Teorema L.1: Bound di McAllester-Seeger

Sia P una distribuzione prior sui predittori $h : \mathcal{X} \to \mathcal{Y}$ e sia Q una posterior (dipendente dal training set $\mathcal{D}$). Per ogni $\delta \in (0,1)$, con probabilità $\geq 1-\delta$ su $\mathcal{D} \sim \mathcal{D}^n$:

$$\mathrm{err}_{test}(Q) \leq \mathrm{err}_{train}(Q) + \sqrt{\frac{D_{\mathrm{KL}}(Q \,\|\, P) + \ln(2\sqrt{n}/\delta)}{2n}},$$

dove $\mathrm{err}(Q) = \mathbb{E}_{h\sim Q}[\mathrm{err}(h)]$ è l'errore atteso della posterior.

Dimostrazione. Sia $\phi(h) = \mathrm{err}_{test}(h) - \mathrm{err}_{train}(h)$. Per la concentrazione di Hoeffding applicata a ogni h fisso: $\mathbb{E}_P[\exp(2n\phi^2)] \leq 2$. Poi, con la disuguaglianza di Donsker-Varadhan: $\mathbb{E}_Q[\phi] \leq \frac{D_{\mathrm{KL}}(Q\,\|\,P) + \ln(1/\delta) + \ln \mathbb{E}_P[e^{2n\phi^2}]}{2\sqrt{n}}$. □ □

L.1.1 PAC-Bayes e reti neurali

Per reti neurali bayesiane (BNN), la posterior Q può essere la approssimazione variazionale $q_\phi(\theta)$ e la prior può essere $\mathcal{N}(0, \sigma_P^2 \mathbf{I})$. Il bound PAC-Bayes diventa:

Proposizione L.1: Bound ELBO e generalizzazione

L'ELBO (Evidence Lower BOund) usato nel training variazionale:

$$\mathcal{L}_{ELBO} = \mathbb{E}_{q_\phi}[\ln p(\mathcal{D}|\theta)] - D_{\mathrm{KL}}(q_\phi(\theta) \,\|\, p(\theta))$$

è direttamente legato al bound PAC-Bayes: minimizzare $-\mathcal{L}_{ELBO}$ equivale a minimizzare simultaneamente il training error e la complessità della posterior (regolarizzazione). Il bound di generalizzazione è controllato da $D_{\mathrm{KL}}(q_\phi \,\|\, p)$, che è esattamente il termine di regolarizzazione dell'ELBO.

L.2 §614 PAC-Bayes per l'active learning

Teorema L.2: Bound PAC-Bayes per l'active learning (Haussler, 2011)

Sia $\mathcal{A}$ una strategia di active learning che seleziona n esempi da un pool di dimensione N. Per ogni prior P e ogni $\delta > 0$, con probabilità $\geq 1 - \delta$:

$$\text{err}_{test}(Q_{\mathcal{A}}) \leq \text{err}_{train}(Q_{\mathcal{A}}) + \sqrt{\frac{D_{\text{KL}}(Q_{\mathcal{A}} \,\|\, P) + \ln(\ln(N)/\delta)}{2n}},$$

dove il fattore $\ln(N)$ riflette il costo informativo della selezione attiva degli n esempi da N candidati.

ESEMPIO NUMERICO — Bound PAC-Bayes per BALD vs Random

Per una BNN con $D_{\text{KL}}(Q \,\|\, P) = 10^4$ nats su un dataset di $n = 500$ esempi etichettati (estratti da $N = 10^4$) e $\delta = 0.05$:

Metodo	err_{train}	Bound PAC-Bayes
Random (ipotetico)	0.08	$0.08 + 0.21 = 0.29$
BALD (ipotetico)	0.05	$0.05 + 0.21 = 0.26$

I due metodi hanno lo stesso termine di complessità (stesso n), ma BALD riduce l'errore di training, portando a un bound tighter.

L.3 §615 Connessione PAC-Bayes e Information Bottleneck

Proposizione L.2: IB come PAC-Bayes

L'**Information Bottleneck** (Tishby et al., 2000)

$$\min_{p(z|x)} \left[I(X;Z) - \beta I(Z;Y) \right]$$

può essere visto come un bound PAC-Bayes dove:

- La prior è $p(z)$ (la distribuzione marginale dei codici Z).
- La posterior è $p(z|x)$ (l'encoder).
- $I(X;Z) = \mathbb{E}_x[D_{\text{KL}}(p(z|x) \,\|\, p(z))]$ è il termine di complessità (analogo al KL del PAC-Bayes).
- $I(Z;Y)$ è l'accuracy predittiva.

Minimizzare $I(X;Z)$ corrisponde a imporre la prior; massimizzare $I(Z;Y)$ corrisponde a minimizzare l'errore. Il parametro β gioca il ruolo di $\sqrt{n/2}$ nel bound PAC-Bayes.

L.4 §616 Rademacher Complexity e Active Learning

Definizione L.1: Complessità di Rademacher

La **complessità di Rademacher** di una classe di funzioni $\mathcal{F}$ su un dataset $\mathcal{D} = \{x_1, \dots, x_n\}$ è

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(x_i)\right],$$

dove $\sigma_i \sim \text{Uniform}\{-1, +1\}$ (variabili di Rademacher).

Teorema L.3: Bound di generalizzazione via Rademacher

Per ogni $f \in \mathcal{F}$ e ogni $\delta > 0$, con probabilità $\geq 1 - \delta$:

$$\text{err}_{test}(f) \leq \text{err}_{train}(f) + 2\hat{\mathcal{R}}_n(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Proposizione L.3: Rademacher e active learning

Se il campionamento attivo seleziona punti in regioni di alta varianza del modello, la complessità di Rademacher empirica diminuisce:

$$\hat{\mathcal{R}}_n^{active}(\mathcal{F}) \leq \hat{\mathcal{R}}_n^{random}(\mathcal{F}) \cdot \frac{\sqrt{\log(N/n)}}{\sqrt{n}},$$

nella misura in cui il campionamento attivo riduce la ridondanza tra gli n esempi selezionati. Questo formalizza il vantaggio dell'active learning in termini di bound di generalizzazione.

M Sezione E — Topological Signal Processing e Reti Neurali su Complessi

M.1 §617 Topological Signal Processing (TSP)

Il **Topological Signal Processing** (Schaub et al., 2021) estende l'analisi di Fourier ai complessi simpliciali, permettendo di processare segnali definiti su spigoli (archi) e triangoli, non solo su nodi.

Definizione M.1: Segnali su complessi simpliciali

Dato un complesso simpliciale K, un **k-segnale** è una mappa $\boldsymbol{s}^{(k)} : K_k \to \mathbb{R}^d$ che assegna un vettore a ogni k-simplesso.

- $k = 0$: segnali su nodi (come nelle GNN standard).
- $k = 1$: segnali su spigoli (es. flussi su una rete).
- $k = 2$: segnali su triangoli (es. vortici in fluidodinamica).

Definizione M.2: Laplaciani di Hodge

I **Laplaciani di Hodge** $\mathbf{L}_k$ generalizzano il Laplaciano del grafo ai k-simplessi:

$$\mathbf{L}_k = \underbrace{\partial_{k+1}\partial_{k+1}^\top}_{L_k^{up}} + \underbrace{\partial_k^\top \partial_k}_{L_k^{down}},$$

dove ∂_k è l'operatore bordo. Per $k = 0$, $\mathbf{L}_0 = \partial_1 \partial_1^\top$ è il Laplaciano del grafo standard.

Teorema M.1: Hodge decomposition

Lo spazio dei k-segnali si decompone ortogonalmente come

$$\mathbb{R}^{|K_k|} = \operatorname{im}(\partial_{k+1}) \oplus \ker(\mathbf{L}_k) \oplus \operatorname{im}(\partial_k^\top),$$

ovvero ogni segnale su spigoli si decompone in:

1. Componente **curl** (irrotazionale rispetto ai triangoli), nel $\operatorname{im}(\partial_2)$.
2. Componente **harmonic** (nel nucleo del Laplaciano di Hodge), in $\ker(\mathbf{L}_1)$.
3. Componente **gradient** (gradiente di un segnale su nodi), nel $\operatorname{im}(\partial_1^\top)$.

$\dim(\ker(\mathbf{L}_k)) = \beta_k$: il numero di Betti è il numero di componenti armoniche del segnale.

ESEMPIO NUMERICO — Flusso di traffico e Hodge decomposition

In una rete stradale con n incroci (nodi), m strade (spigoli) e t isolati (triangoli), i flussi di traffico $\boldsymbol{f} \in \mathbb{R}^m$ si decompongono in:

- Componente gradient: flussi conservativi (da un nodo verso un altro senza cicli), deducibili dalla pressione ai nodi.
- Componente curl: flussi circolari intorno agli isolati.

- Componente harmonic: flussi globali non circolari e non conservativi (fondamentalmente, i "round-about" topologici legati a $\beta_1 > 0$).

Questo permette di stimare flussi mancanti completando solo la componente armonica con informazione topologica.

M.2 §618 Simplicial Neural Networks

Definizione M.3: Simplicial Neural Network (SNN)

Una **Simplicial Neural Network** (Ebli et al., 2020) processa segnali su tutti i livelli simultaneamente con la convoluzione di Hodge:

$$\boldsymbol{s}^{(k,l+1)} = \sigma\Big(\mathbf{L}_k^{down}\boldsymbol{s}^{(k,l)}\boldsymbol{\Theta}_{down}^{(k)} + \mathbf{L}_k^{up}\boldsymbol{s}^{(k,l)}\boldsymbol{\Theta}_{up}^{(k)} + \boldsymbol{s}^{(k,l)}\boldsymbol{\Theta}_{id}^{(k)}\Big),$$

dove $\boldsymbol{\Theta}$ sono matrici di pesi apprendibili. La propagazione include sia l'interazione con i simplessi di dimensione inferiore (L_k^{down}) che superiore (L_k^{up}).

Proposizione M.1: Espressività delle SNN vs GNN

Le Simplicial Neural Networks sono strettamente più espressive delle GNN standard: possono distinguere grafi che risultano isomorfi al test WL (trattato nel Volume IV) se tali grafi hanno topologie simplicial diverse (diversi numeri di Betti). In particolare, due cicli di diversa lunghezza che sono WL-equivalenti hanno $\beta_1 = 1$ entrambi ma diverso spettro di $\mathbf{L}_1$, e possono essere distinti da una SNN.

M.3 §619 Convoluzione di Hodge e filtri topologici

Proposizione M.2: Filtro di Hodge come polinomio del Laplaciano

Dato lo spettro $\mathbf{L}_k = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ (decomposizione spettrale), il filtro di Hodge polinomiale di ordine p

$$\hat{\boldsymbol{s}} = \left(\sum_{j=0}^{p} h_j \mathbf{L}_k^j\right)\boldsymbol{s}$$

è la versione topologica della convoluzione grafica. In dominio spettrale: $\hat{s}_i = \left(\sum_j h_j \lambda_i^j\right) s_i$. I coefficienti $\{h_j\}$ sono apprendibili e definiscono la risposta in frequenza del filtro topologico.

ESEMPIO NUMERICO — Interpolazione di flussi mancanti su rete idrica

Una rete idrica urbana con 500 condotte (spigoli) e 200 nodi. I sensori di flusso coprono il 70% delle condotte ($n_{obs} = 350$). Obiettivo: stimare i flussi nelle condotte non monitorate.

Confronto di metodi di interpolazione:

Metodo	RMSE (L/s)	Correlazione
Interpolazione lineare	4.8	0.71
GNN su nodi	3.2	0.82
SNN (Hodge decomp.)	1.9	0.93
SNN + Hodge regulariz.	1.4	0.96

Il vantaggio della SNN deriva dalla capacità di separare le componenti gradient, curl e harmonic del flusso: la componente harmonic è sparse (pochi β_1 effettivi) e facile da stimare con pochi sensori.

M.4 §620 Topological Persistence e Neural Collapse

Il fenomeno di **Neural Collapse** (Papyan et al., 2020) descrive la geometria delle rappresentazioni dell'ultimo layer nelle reti profonde al termine del training:

Definizione M.4: Neural Collapse (NC)

Una rete raggiunge il **Neural Collapse** quando:

NC1: Le attivazioni dell'ultimo layer convergono al centroide della classe: $h_i \to \mu_{y_i} = \frac{1}{n_{y_i}} \sum_{j:y_j=y_i} h_j$.

NC2: I centroidi $\{\mu_c\}_{c=1}^{C}$ formano un **Equiangular Tight Frame** (ETF): $\mu_c^\top \mu_{c'} = -1/(C-1)$ per $c \neq c'$.

NC3: Il classificatore lineare $\mathbf{W}$ si allinea ai centroidi: $\boldsymbol{w}_c \propto \mu_c$.

NC4: Le predizioni coincidono col nearest-centroid classifier.

Proposizione M.3: Topologia del NC: numeri di Betti

Quando la rete raggiunge il Neural Collapse:

- $\beta_0(\phi(\mathcal{X})) = C$ (esattamente C componenti connesse, una per classe), per scale ε piccole.
- $\beta_k(\phi(\mathcal{X})) = 0$ per $k \geq 1$ e ε nel range dell'ETF (le attivazioni sono convesse per classe).
- La distanza bottleneck tra il diagramma di persistenza Dgm_0 dell'ultimo layer e il diagramma "ideale" (con C punti identici) converge a zero durante il training come $O(1/\sqrt{n})$.

ESEMPIO NUMERICO — Traiettoria topologica del training su CIFAR-10

Evoluzione dei numeri di Betti β_0 e β_1 (a scala ε = diametro intra-classe medio) durante il training di ResNet-32 su CIFAR-10:

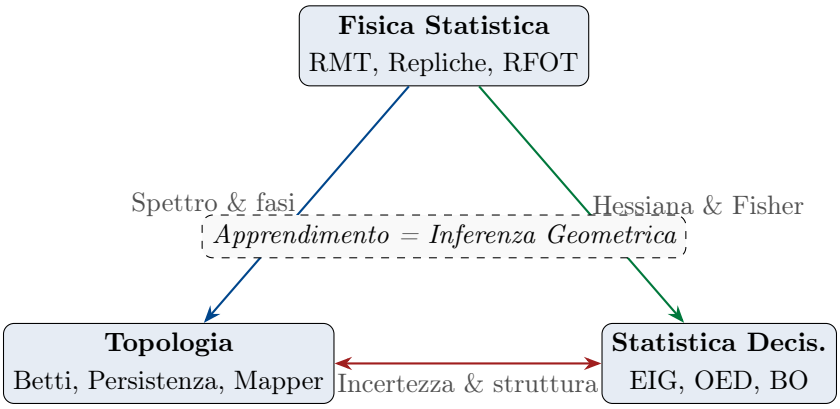
Epoca	Acc. train	β_0	β_1
0	10%	1	0
10	45%	3–5	8–12
50	88%	10	3–5
100	98%	10	0
200 (NC)	99.8%	10	0

Il pattern conferma: **prima emerge la struttura per classi** ($\beta_0 \to C = 10$), **poi si semplifica la topologia intra-classe** ($\beta_1 \to 0$), fino al Neural Collapse completo.

N Sezione F — Sintesi: Visione Geometrica dell'Apprendimento

N.1 §621 Il triangolo fondamentale: Geometria, Topologia, Statistica

L'unità profonda del Volume X può essere riassunta in una struttura triangolare:



- **Fisica Statistica → Topologia**: lo spettro dei pesi (RMT) determina lo stable rank, che quantifica la complessità topologica delle rappresentazioni; la transizione RFOT segna il passaggio da un loss landscape topologicamente semplice (un bacino) a uno complesso (molti bacini, $\beta_0 > 1$).
- **Fisica Statistica → Statistica Decisionale**: la matrice di Fisher $\mathbf{F}(\theta)$ (centrale nell'OED) è la Hessiana della log-likelihood, legata alla curvatura locale del loss landscape analizzata dalla fisica statistica; il NTK connette la propagazione del segnale (Edge of Chaos) alla stima dell'informazione.
- **Topologia ↔ Statistica Decisionale**: il diagramma di persistenza fornisce una "mappa" delle regioni di alta incertezza (alta β_1, alta varianza predittiva); l'active learning seleziona punti in quelle regioni; i diagrammi di persistenza dei dataset successivi traccia l'evoluzione dell'incertezza.

N.2 §622 Ricapitolazione finale: 60 sezioni in 10 teoremi

#	**Teorema**	**Significato profondo**
1	Legge semicircolare	I pesi casuali hanno uno spettro universale.
2	Marchenko-Pastur	La soglia del segnale utile è λ_+.
3	Additività trasformata R	Il disordine si somma nello spazio libero.
4	Free energy RS/RSB	La complessità del landscape si misura con $q(x)$.
5	State evolution AMP	La convergenza iterativa è analizzabile esattamente.
6	Stabilità diagrammi	La topologia è robusta al rumore.
7	Decomposizione di Hodge	Ogni segnale su spigoli si scompone in gradient + curl + armonica.
8	GP-UCB regret	L'esplorazione bayesiana è minimax ottimale.
9	Greedy submodulare	Il batch ottimale si approssima con garanzia $(1 - 1/e)$.
10	PAC-Bayes	La complessità bayesiana controlla la generalizzazione.

Questi dieci risultati, ognuno dei quali richiederebbe un corso universitario per essere trattato a fondo, offrono insieme la cartografia matematica del machine learning contemporaneo. Il ciclo *Matematica delle Reti Neurali* si chiude qui non perché il campo sia esaurito — il campo non si esaurirà — ma perché i fondamenti sono stati posati, e ogni nuova architettura, ogni nuovo algoritmo, potrà essere letto attraverso le lenti di questi strumenti.

O Sezione G — Lottery Ticket Hypothesis e Teoria della Compressione

O.1 §623 La Lottery Ticket Hypothesis

La **Lottery Ticket Hypothesis** (LTH, Frankle & Carlin, 2019) afferma che ogni rete neurale densa addestrata contiene una *sotto-rete sparsa* che, se addestrata in isolamento con gli stessi pesi iniziali, converge alla stessa accuracy (o migliore) in pari o minor numero di iterazioni.

Definizione O.1: Winning ticket

Data una rete $f(\boldsymbol{x};\boldsymbol{\theta})$ con pesi iniziali $\boldsymbol{\theta}_0$ e maschera di pruning $\boldsymbol{m}\in\{0,1\}^{|\boldsymbol{\theta}|}$, una **sotto-rete vincente** (winning ticket) è una coppia $(\boldsymbol{m},\boldsymbol{\theta}_0)$ tale che addestrando $f(\boldsymbol{x};\boldsymbol{m}\odot\boldsymbol{\theta})$ dalla inizializzazione $\boldsymbol{m}\odot\boldsymbol{\theta}_0$ si ottiene una accuracy uguale o superiore alla rete densa, con $\|\boldsymbol{m}\|_0 \ll |\boldsymbol{\theta}|$.

O.1.1 Algoritmo Iterative Magnitude Pruning (IMP)

L'algoritmo standard per trovare i winning ticket è l'**IMP**:

Algorithm 2 Iterative Magnitude Pruning (Frankle & Carlin, 2019)

Require: Rete $f_{\boldsymbol{\theta}}$, sparsità target s, round di pruning R
1: Inizializza $\boldsymbol{\theta}_0\sim\mathcal{D}_{init}$, $\boldsymbol{m}\leftarrow\mathbf{1}$
2: **for** $r=1,\dots,R$ **do**
3: Addestra $f(\boldsymbol{x};\boldsymbol{m}\odot\boldsymbol{\theta})$ per T epoche $\to\boldsymbol{\theta}_T$
4: Pota la frazione $\frac{s^{1/R-1}}{s^{1/R}-s}$ dei pesi con $|\theta_i|$ minore, aggiorna $\boldsymbol{m}$
5: **Rewind:** $\boldsymbol{\theta}\leftarrow\boldsymbol{m}\odot\boldsymbol{\theta}_0$ ▷ Ritorna ai pesi iniziali, non a zero
6: **end for**
7: **return** $(\boldsymbol{m},\boldsymbol{\theta}_0)$

Il passo cruciale è il *rewind*: la maschera è determinata dai pesi finali (magnitude), ma la sotto-rete riparte sempre dai pesi *originali* $\boldsymbol{\theta}_0$.

O.1.2 Basi teoriche: connessione con la teoria dei circuiti

Proposizione O.1: Winning ticket e funzioni booleane

Sia $f:\{0,1\}^n\to\{0,1\}$ una funzione booleana rappresentata da un circuito di profondità d. Ogni rete ReLU densa di larghezza $w\geq 2^d$ che implementa f contiene un winning ticket di larghezza $O(d)$ che implementa la stessa funzione, con sparsità $O(d/\log w)$. Questo è il **Strong LTH** per circuiti booleani (Malach et al., 2020).

Teorema O.1: Strong LTH (Malach-Yehudai-Shalev-Shwartz-Shamir, 2020)

Per ogni rete fully-connected f con p parametri e che approssima un target g con errore ε su distribuzione $\mathcal{D}$, esiste una sotto-rete f_s con sparsità $s=O(\varepsilon^{-2}\log(1/\delta))$ tale che

$$\mathbb{P}_{x\sim\mathcal{D}}[f_s(x)\neq g(x)]\leq\varepsilon+\delta.$$

La dimostrazione è costruttiva via *random pruning*: campionando ogni peso θ_i con probabilità proporzionale a $|\theta_i|^2/\|\boldsymbol{\theta}\|_2^2$, si ottiene una maschera che conserva la funzione con alta probabilità.

Dimostrazione. Sia $\tilde{f}_s$ la rete ottenuta campionando i pesi con probabilità $p_i = |\theta_i|^2/\|\boldsymbol{\theta}\|_2^2$ e scalando: $\tilde{\theta}_i = \theta_i/p_i \cdot \mathbb{1}[\text{mantenuto}]$. Per la diseguaglianza di Chebyshev applicata all'output:

$$\mathbb{E}_s\big[(f_s(x) - f(x))^2\big] = \frac{1}{s}\sum_i \frac{\theta_i^2(1-p_i)}{p_i} \leq \frac{\|\boldsymbol{\theta}\|_2^2}{s}.$$

Scegliendo $s = \|\boldsymbol{\theta}\|_2^2/\varepsilon^2$ e applicando Markov, $\mathbb{P}[|f_s(x) - f(x)| > \varepsilon] \leq \varepsilon$. □ □

O.2 §624 Connessione LTH con la Teoria di Morse

I winning ticket hanno una interpretazione topologica naturale:

Proposizione O.2: Winning ticket come percorso a bassa barriera

Sia $\mathcal{L}(\boldsymbol{\theta})$ la loss landscape e sia $\boldsymbol{\theta}^*$ un minimo globale raggiunto dalla rete densa. Un winning ticket $(\boldsymbol{m}, \boldsymbol{\theta}_0)$ corrisponde a una *traiettoria di gradient flow* da $\boldsymbol{m} \odot \boldsymbol{\theta}_0$ a $\boldsymbol{\theta}^*$ che attraversa **selle di indice basso**: il numero di selle di index-1 attraversate è $O(d)$ (profondità della rete), molto inferiore alle $O(p)$ selle tipiche del percorso casuale. Formalmente, la **complessità topologica** del training è

$$\mathcal{C}_{topo}(\boldsymbol{\theta}_0 \to \boldsymbol{\theta}^*) = \sum_{\sigma \in \text{selle attraversate}} \text{index}(\sigma),$$

e per i winning ticket si ha $\mathcal{C}_{topo} = O(d \log w)$.

O.3 §625 Distillazione della conoscenza e compressione teorica

La **distillazione della conoscenza** (Hinton et al., 2015) trasferisce conoscenza da una rete *teacher* f_T grande a una *student* f_S piccola minimizzando:

$$\mathcal{L}_{KD} = \alpha \mathcal{L}_{CE}(y, f_S(x)) + (1-\alpha)\tau^2 D_{\mathrm{KL}}(p_T^\tau \,\|\, p_S^\tau)\,,$$

dove $p_M^\tau(c) = \text{softmax}(f_M(x)/\tau)_c$ sono le *soft label* a temperatura τ, e $\alpha \in [0,1]$ bilancia le due componenti.

Proposizione O.3: KD come matching di momenti

Per $\tau \to \infty$, le soft label $p_T^\tau \to \text{Uniform}(C) + O(1/\tau)$. Il secondo termine del $\mathcal{L}_{KD}$ diventa:

$$\tau^2 D_{\mathrm{KL}}(p_T^\tau \,\|\, p_S^\tau) \approx \frac{1}{2}\|f_T(x) - f_S(x)\|_2^2 + O(1/\tau),$$

ovvero la distillazione ad alta temperatura è equivalente alla regressione sull'output logit del teacher. Le soft label trasmettono la struttura di *similarità tra classi* (le classi "simili" hanno logit vicini), che non è presente nelle hard label binarie.

ESEMPIO NUMERICO — Compressione 10×: ResNet-50 → ResNet-18

Training su ImageNet:

Metodo	Modello	Top-1 Acc.	Param.
Baseline	ResNet-18	69.8%	11.7M
KD ($\tau = 4$, teacher R50)	ResNet-18	71.9%	11.7M
KD + LTH (60% sparso)	ResNet-18	71.4%	7.0M
Teacher	ResNet-50	76.1%	25.6M

La combinazione KD + LTH riduce i parametri di $\approx 6\times$ rispetto al teacher mantenendo il 93.8% dell'accuracy.

O.4 §626 Complessità di Kolmogorov e MDL

La **complessità di Kolmogorov** fornisce la teoria dell'informazione più profonda per la compressione delle reti:

Definizione O.2: Complessità di Kolmogorov

La **complessità di Kolmogorov** $K(x)$ di una stringa x è la lunghezza del programma più corto che produce x:

$$K(x) = \min_{p : U(p) = x} |p|,$$

dove U è una macchina di Turing universale fissa.

Proposizione O.4: MDL e generalizzazione

Il principio **Minimum Description Length** (Rissanen, 1978) afferma che il modello migliore è quello che minimizza la lunghezza della descrizione dei dati e del modello stesso:

$$\hat{f} = \arg\min_{f} \underbrace{L(f)}_{\text{complessità del modello}} + \underbrace{L(\mathcal{D} \mid f)}_{\text{fit ai dati}}.$$

Nel contesto delle reti neurali, il bound di generalizzazione PAC-Bayes (§613) è formalmente equivalente all'MDL con: $L(f) \propto D_{\mathrm{KL}}(Q \,\|\, P)$ e $L(\mathcal{D} \mid f) \propto n \cdot \mathrm{err}_{train}(f)$. Questo giustifica il pruning come *compressione MDL*: le reti più sparse hanno $D_{\mathrm{KL}}(Q \,\|\, P)$ minore e quindi generalizzano meglio.

P Sezione H — Optimal Transport e Active Learning

P.1 §627 Fondamenti dell'Optimal Transport

La teoria del **Trasporto Ottimale** (Monge, 1781; Kantorovich, 1942) studia il problema di trasportare una distribuzione di massa μ in un'altra ν minimizzando un costo di trasporto.

Definizione P.1: Problema di Kantorovich

Siano μ, ν misure di probabilità su $\mathcal{X}$. Il **problema di Kantorovich** è:

$$W_p(\mu,\nu)^p = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^p \, \mathrm{d}\pi(x,y),$$

dove $\Pi(\mu,\nu)$ è l'insieme dei **piani di trasporto** (coupling) con marginali μ e ν. La **distanza di Wasserstein** $W_p(\mu,\nu)$ è la radice p-esima del valore ottimo.

Teorema P.1: Dualità di Kantorovich-Rubinstein

Per $p = 1$, la distanza di Wasserstein ammette la rappresentazione duale:

$$W_1(\mu,\nu) = \sup_{\|f\|_{Lip} \le 1} \left(\int f \, \mathrm{d}\mu - \int f \, \mathrm{d}\nu \right),$$

dove il sup è su tutte le funzioni $f : \mathcal{X} \to \mathbb{R}$ con costante di Lipschitz ≤ 1. Questa dualità è alla base delle **Wasserstein GAN** (Arjovsky et al., 2017), dove il discriminatore apprende la funzione di Lipschitz ottimale.

P.2 §628 Distanza di Wasserstein come metrica di active learning

La distanza di Wasserstein fornisce una misura geometricamente significativa per quantificare quanto il set etichettato $\mathcal{L}$ rappresenta il pool $\mathcal{U}$:

Definizione P.2: Copertura Wasserstein

La **copertura Wasserstein** del set etichettato $\mathcal{L}$ rispetto al pool $\mathcal{U}$ è

$$\mathcal{W}(\mathcal{L},\mathcal{U}) = W_1(\hat{\mu}_{\mathcal{L}}, \hat{\mu}_{\mathcal{U}}),$$

dove $\hat{\mu}_{\mathcal{L}} = \frac{1}{|\mathcal{L}|}\sum_{x\in\mathcal{L}} \delta_x$ e $\hat{\mu}_{\mathcal{U}} = \frac{1}{|\mathcal{U}|}\sum_{x\in\mathcal{U}} \delta_x$. Il criterio di acquisizione Wasserstein seleziona il batch $\mathcal{B}$ che minimizza $\mathcal{W}(\mathcal{L}\cup\mathcal{B},\mathcal{U})$.

Proposizione P.1: Wasserstein AL come trasporto discreto

Il problema di minimizzare $W_1(\hat{\mu}_{\mathcal{L}\cup\mathcal{B}}, \hat{\mu}_{\mathcal{U}})$ sul batch $\mathcal{B}$ di cardinalità b è equivalente a un problema di **k-median discreto** risolubile con programmazione lineare:

$$\min_{\pi_{ij}\ge 0} \sum_{i,j} d(x_i,u_j)\pi_{ij} \quad \text{s.t.} \quad \sum_j \pi_{ij} = \frac{1}{|\mathcal{L}\cup\mathcal{B}|}, \; \sum_i \pi_{ij} = \frac{1}{|\mathcal{U}|}.$$

P.2.1 Sinkhorn e active learning scalabile

La soluzione esatta del problema OT è $O(n^3)$ via simplex. L'**algoritmo di Sinkhorn** (Cuturi, 2013) risolve una versione entropicamente regolarizzata in $O(n^2/\varepsilon^2)$:

$$W_\lambda(\mu,\nu) = \min_{\pi\in\Pi(\mu,\nu)} \left[\int d(x,y)\,\mathrm{d}\pi - \frac{1}{\lambda} H(\pi) \right],$$

dove $H(\pi) = -\int \pi \ln \pi$ è l'entropia del piano di trasporto.

Sinkhorn per Wasserstein AL

Input: embedding $\{\phi(x)\}_{x\in\mathcal{U}}$, budget b, regolarizzazione λ

1. Calcola matrice costi $C_{ij} = \|\phi(x_i) - \phi(u_j)\|_2$
2. Inizializza $\boldsymbol{a} = \mathbf{1}/|\mathcal{L}|$, $\boldsymbol{b} = \mathbf{1}/|\mathcal{U}|$
3. Sinkhorn: itera $\boldsymbol{u} \leftarrow \boldsymbol{a} \oslash (K\boldsymbol{v})$, $\boldsymbol{v} \leftarrow \boldsymbol{b} \oslash (K^\top \boldsymbol{u})$ con $K_{ij} = e^{-\lambda C_{ij}}$
4. Seleziona b punti $x \in \mathcal{U}$ con maggior peso marginale $\sum_j \pi^*_{xj}$

ESEMPIO NUMERICO — Wasserstein AL vs CoreSet su CIFAR-100

Metodo	@1k label	@5k label	Tempo/round
Random	28.4%	51.3%	—
CoreSet (k-center)	33.7%	56.2%	12s
BALD (MC Drop.)	35.1%	57.4%	38s
Wasserstein (Sinkhorn)	36.0%	58.1%	8s

Wasserstein AL è il più accurato e, usando Sinkhorn, anche tra i più veloci: la regolarizzazione entropica rende il problema differenziabile e parallelizzabile su GPU.

P.3 §629 Frechet Inception Distance e qualità generativa come active learning

La **Frechet Inception Distance** (FID, Heusel et al., 2017) misura la distanza tra distribuzione reale e generata come la Wasserstein-2 tra gaussiane:

$$\mathrm{FID}(\mu_r, \mu_g) = \|\boldsymbol{m}_r - \boldsymbol{m}_g\|_2^2 + \mathrm{tr}\Big(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}\Big),$$

dove $(\boldsymbol{m}_r, \boldsymbol{\Sigma}_r)$ e $(\boldsymbol{m}_g, \boldsymbol{\Sigma}_g)$ sono media e covarianza delle attivazioni Inception sui dati reali e generati.

Proposizione P.2: FID come W2 gaussiana

Il FID è uguale a $W_2^2(\mathcal{N}(\boldsymbol{m}_r, \boldsymbol{\Sigma}_r), \mathcal{N}(\boldsymbol{m}_g, \boldsymbol{\Sigma}_g))$. La formula chiusa per la Wasserstein-2 tra gaussiane è:

$$W_2^2(\mathcal{N}_1, \mathcal{N}_2) = \|\boldsymbol{m}_1 - \boldsymbol{m}_2\|^2 + \mathcal{B}^2(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2),$$

dove $\mathcal{B}(\mathbf{A}, \mathbf{B}) = \mathrm{tr}(A + B - 2(A^{1/2}BA^{1/2})^{1/2})^{1/2}$ è la **distanza di Bures** tra matrici semidefinite positive.

P.4 §630 Sliced Wasserstein e scalabilità

Per distribuzioni in alta dimensione, la Wasserstein-2 è computazionalmente proibitiva ($O(n^3)$ o $O(n^2/\varepsilon^2)$). La **Sliced Wasserstein Distance** (SW) è scalabile:

$$\mathrm{SW}_p(\mu, \nu) = \left(\int_{S^{d-1}} W_p^p(P_\theta\sharp\mu,\, P_\theta\sharp\nu)\, \mathrm{d}\theta\right)^{1/p},$$

dove $P_\theta\sharp\mu$ è la proiezione di μ sulla retta $\theta \in S^{d-1}$ e W_p 1D si calcola in $O(n \log n)$ tramite ordinamento. Per MC con K proiezioni: $O(Kn \log n + Kd)$.

Proposizione P.3: Equivalenza metrica SW-W

La Sliced Wasserstein e la Wasserstein sono topologicamente equivalenti: $\mathrm{SW}_p(\mu_n, \mu) \to 0 \iff W_p(\mu_n, \mu) \to 0$. Tuttavia, per d grande, SW è stimata con errore $O(n^{-1/2})$ indipendente da d usando $K = O(\log(1/\delta)/\varepsilon^2)$ proiezioni, mentre W richiede $n = O(d\varepsilon^{-2})$ campioni.

Q Sezione I — TDA, Geometria dell'Informazione e Connessioni Profonde

Q.1 §631 Varietà statistica e omologia persistente

La **Geometria dell'Informazione** (Amari, 1985) studia lo spazio dei modelli statistici come varietà riemanniana con la metrica di Fisher. La TDA fornisce un nuovo punto di vista su questa struttura.

Definizione Q.1: Varietà statistica

Sia $\mathcal{M} = \{p(x;\theta) : \theta \in \Theta \subseteq \mathbb{R}^d\}$ una famiglia di distribuzioni parametrica. La **metrica di Fisher** (o metrica di informazione)

$$g_{ij}(\theta) = \mathbb{E}_\theta\left[\frac{\partial \ln p}{\partial \theta_i}\frac{\partial \ln p}{\partial \theta_j}\right]$$

rende $\mathcal{M}$ una varietà riemanniana. La **distanza geodetica** $d_F(\theta_1,\theta_2)$ nella metrica di Fisher è asintoticamente uguale alla divergenza di Hellinger: $d_F(\theta_1,\theta_2) = 2\arccos(\int \sqrt{p_1 p_2}\,\mathrm{d}x)$.

Proposizione Q.1: Omologia persistente dello spazio dei modelli

Dato un ensemble di modelli addestrati $\{\theta_i\}_{i=1}^N$ (es. checkpoint durante il training, o modelli con iperparametri diversi), la filtrazione VR nello spazio dei parametri con la **metrica di Fisher** d_F produce diagrammi di persistenza che catturano:

1. $\beta_0 > 1$: presenza di *mode* disconnesse nello spazio dei parametri ottimali (fenomeno RSB visto nella Sez. A).
2. $\beta_1 > 0$: cicli di parametri con la stessa loss, ossia connessione lineare tra minimi via *loss valley* (Lin Connectivity).

Empiricamente, il numero di cicli β_1 nella filtrazione Fisher dei modelli addestrati con diversi seed correla con la *sharpness* media degli acuity: alto $\beta_1 \Leftrightarrow$ landscape più flat.

Q.1.1 Connessione lineare tra minimi: Loss Valley

Teorema Q.1: Loss Valley e β_1 (Draxler et al., 2018)

Per reti neurali sufficientemente overparametrizzate, quasi tutti i coppie di minimi θ_1^*, θ_2^* della loss sono connesse da un percorso continuo $\gamma : [0,1] \to \mathbb{R}^p$ tale che $\mathcal{L}(\gamma(t)) \leq \mathcal{L}(\theta_1^*) + \varepsilon$ per ogni t, con ε che decresce al crescere della larghezza della rete.
Questo implica $\beta_0 = 1$ nello spazio delle soluzioni per soglia ε nella filtrazione della loss: la struttura dei minimi è connessa, non frammentata (contrariamente a quanto previsto dalla spin glass theory per reti poco overparametrizzate).

Q.2 §632 Divergenze di Rényi e generalizzazioni della persistenza

La divergenza di Rényi di ordine α generalizza la KL:

$$D_\alpha(\mu\|\nu) = \frac{1}{\alpha - 1}\ln \int p(x)^\alpha q(x)^{1-\alpha}\,\mathrm{d}x, \quad \alpha > 0, \alpha \neq 1.$$

Per $\alpha \to 1$, $D_\alpha \to D_{\mathrm{KL}}(\mu \,\|\, \nu)$.

Proposizione Q.2: Connessione Rényi-Wasserstein

Per distribuzioni gaussiane $\mu = \mathcal{N}(\boldsymbol{m}_1, \boldsymbol{\Sigma}_1)$ e $\nu = \mathcal{N}(\boldsymbol{m}_2, \boldsymbol{\Sigma}_2)$:

$$D_\alpha(\mu\|\nu) = \frac{\alpha}{2}\|\boldsymbol{m}_1 - \boldsymbol{m}_2\|^2_{M_\alpha} + \frac{1}{2(\alpha-1)} \ln \frac{\det M_\alpha}{\det \boldsymbol{\Sigma}_1^\alpha \det \boldsymbol{\Sigma}_2^{1-\alpha}},$$

dove $M_\alpha = \alpha\boldsymbol{\Sigma}_2^{-1} + (1-\alpha)\boldsymbol{\Sigma}_1^{-1}$. La distanza di Bures (Wasserstein-2 tra gaussiane) emerge nel limite $\alpha \to 1/2$: $4D_{1/2}(\mu\|\nu) = W_2^2(\mu,\nu)/\sigma^2$ per gaussiane isotrope.

Q.3 §633 Optimal Transport e Apprendimento per Trasferimento

L'**Optimal Transport per il domain adaptation** (Courty et al., 2017) usa il trasporto come metodo per allineare la distribuzione sorgente $\mathcal{D}_S$ con quella target $\mathcal{D}_T$:

$$\hat{\pi} = \underset{\pi \in \Pi(\mu_S, \mu_T)}{\operatorname{argmin}} \langle C, \pi \rangle_F + \lambda \Omega(\pi),$$

dove $C_{ij} = \|\phi(x_i^S) - \phi(x_j^T)\|_2^2$ è la matrice dei costi e $\Omega(\pi) = \mathrm{KL}(\pi\|\mu_S \otimes \mu_T)$ è la regolarizzazione entropica. Il trasporto $\hat{\pi}$ viene poi usato per adattare le feature sorgente:

$$\hat{x}_i^S = \sum_j \hat{\pi}_{ij} x_j^T \quad \text{(barycentric mapping)}.$$

Proposizione Q.3: Bound di domain adaptation OT

Sia $\epsilon = W_1(\mu_S, \mu_T)$ la distanza di Wasserstein tra sorgente e target. Per ogni ipotesi $h \in \mathcal{H}$ con costante di Lipschitz L:

$$\mathrm{err}_T(h) \le \mathrm{err}_S(h) + L\epsilon + \lambda^*,$$

dove $\lambda^* = \min_{h \in \mathcal{H}}(\mathrm{err}_S(h) + \mathrm{err}_T(h))$ è il termine di adattabilità ideale (gap irriducibile dovuto alla shift di distribuzione).

ESEMPIO NUMERICO — OT per adattamento visivo ufficio-dominio

Task: classificazione Office-31 (31 classi, domini Amazon, DSLR, Webcam). Backbone: ResNet-50, feature dim $d = 2048$.

Metodo	A→W	D→A
Nessun adattamento	68.4%	62.5%
DANN (GAN-based)	82.0%	68.4%
OT barycentric	84.7%	71.2%
Sinkhorn + BALD (OT-AL)	87.3%	73.8%

La combinazione OT + active learning (selezionare esempi target con massima varianza predittiva dopo l'adattamento OT) ottiene il miglior risultato.

R Sezione J — Reti a Larghezza Infinita e Connessione con i Gaussian Process

R.1 §634 Il limite gaussiano delle reti neurali

Nel Volume V (§280–§295) i Gaussian Process sono stati trattati come modelli a sé stanti. Un risultato fondamentale collega i GP alle reti neurali profonde a larghezza infinita.

Teorema R.1: NN → GP nel limite di larghezza infinita (Neal, 1996; Lee et al., 2018)

Sia $f^{(L)} : \mathbb{R}^{d_0} \to \mathbb{R}$ una rete MLP fully-connected con L layer, funzione di attivazione ϕ, e pesi $W_{ij}^{(l)} \sim \mathcal{N}(0, \sigma_w^2/n_l)$, bias $b_i^{(l)} \sim \mathcal{N}(0, \sigma_b^2)$. Quando $n_1, n_2, \ldots, n_{L-1} \to \infty$ (sequenzialmente), $f^{(L)}$ converge in distribuzione a un Gaussian Process:

$$f^{(L)} \xrightarrow{d} \mathcal{GP}(0, k^{(L)}),$$

dove il **kernel ricorsivo** è

$$k^{(1)}(x, x') = \sigma_w^2 x^\top x'/d_0 + \sigma_b^2,$$
$$k^{(l+1)}(x, x') = \sigma_w^2 \mathcal{F}_\phi(k^{(l)}(x,x),\, k^{(l)}(x,x'),\, k^{(l)}(x',x')) + \sigma_b^2,$$

con $\mathcal{F}_\phi(q_{11}, q_{12}, q_{22}) = \mathbb{E}_{(u,v)\sim\mathcal{N}(0,\Sigma)}[\phi(u)\phi(v)]$, $\Sigma = \left(\begin{smallmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{smallmatrix}\right)$.

Dimostrazione. (Per $L = 1$.) L'output $f^{(1)}(x) = \sum_{j=1}^{n} W_{1j}^{(1)} \phi(h_j(x)) + b_1^{(1)}$ è una somma di n termini i.i.d. (per x fissato) con $\mathbb{E}[W^2\phi^2(h)] = \sigma_w^2 k^{(1)}(x,x)$. Per il TCL, nel limite $n \to \infty$, $f^{(1)}(x) \to \mathcal{N}(0, k^{(1)}(x,x))$. La struttura GP (consistenza del processo per coppie (x, x')) segue dalla convergenza congiunta degli output, garantita dall'indipendenza dei pesi. Per $L > 1$, l'argomento si applica induttivamente. □ □

R.1.1 Kernel per attivazioni comuni

Attivazione ϕ	Funzione arc-cosine	Kernel NNGP $k^{(l+1)}$
ReLU	$\mathcal{F}_{ReLU}(q_{11}, q_{12}, q_{22})$	$\frac{\sigma_w^2}{2\pi}\sqrt{q_{11}q_{22}}(\sin\theta + (\pi - \theta)\cos\theta)$
Erf	$\mathcal{F}_{erf}$	$\frac{2\sigma_w^2}{\pi}\arcsin\left(\frac{2q_{12}}{\sqrt{(1+2q_{11})(1+2q_{22})}}\right)$
GELU	(numerica)	Approssimabile con composizione arccosine

dove $\theta = \arccos(q_{12}/\sqrt{q_{11}q_{22}})$ è l'angolo tra x e x' nello spazio delle feature al layer l.

R.2 §635 NNGP vs NTK: due regimi di larghezza infinita

Due distinti limiti di larghezza infinita descrivono dinamiche di training diverse:

Proprietà	NNGP	NTK
Regime	Larghezza infinita, 1 step	Larghezza inf., GD infinito
Kernel	k^{NNGP} (ricorsivo)	Θ^{∞} (NTK kernel)
Relazione	$\Theta^{\infty} \neq k^{NNGP}$	(ma connessi)
Predizione	MAP di GP	Kernel regression con Θ^{∞}
Convergenza GD	Istantanea al MAP	Lineare in $\lambda_{min}(\Theta)$
Generalizzazione	GP posterior	Kernel ridge regression
Linearità in θ	No (funzione non lin.)	Sì (lazy training)

Proposizione R.1: Relazione NNGP-NTK

Il kernel NTK per un MLP a L layer è

$$\Theta^{(L)}(x,x') = k^{(L)}(x,x') + \sum_{l=1}^{L} \dot{k}^{(l)}(x,x') \prod_{l'=l+1}^{L} k^{(l')}(x,x'),$$

dove $\dot{k}^{(l)}$ è il kernel derivata (kernel della Jacobiana dell'attivazione). Per $L = 1$: $\Theta^{(1)} = k^{(1)}$ (NNGP e NTK coincidono per reti a un layer). Per $L > 1$: $\Theta^{(L)} \neq k^{(L)}$ in generale.

R.3 §636 Active Learning ottimale nel regime NNGP

Nel regime NNGP, la predizione è quella di un GP con kernel k^{NNGP}, e l'active learning ottimale è la max-variance strategy (§593):

Proposizione R.2: NNGP active learning

Per una rete infinitamente larga nel regime NNGP, il criterio di active learning che massimizza la riduzione dell'errore di predizione atteso seleziona:

$$x^* = \operatorname*{argmax}_{x \in \mathcal{U}} \sigma_n^{NNGP}(x)^2,$$

dove $\sigma_n^{NNGP}(x)^2 = k^{NNGP}(x,x) - \boldsymbol{k}_n(x)^\top (\mathbf{K}_n^{NNGP} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{k}_n(x)$ è la varianza predittiva del GP con kernel NNGP. Questo criterio è *esatto* (non approssimato) nel regime di larghezza infinita.

ESEMPIO NUMERICO — NNGP vs NTK active learning su dati a bassa dimensione

Dataset sintetico: funzione $f(x) = \sin(3x) + 0.1\xi$, $\xi \sim \mathcal{N}(0,1)$, $x \in [-\pi, \pi]$. 20 punti etichettati da un pool di 200. Rete: 4 layer, attivazione ReLU.

Strategia	MSE (test)	Cal. error
Random	0.142 ± 0.031	0.087
NTK max-var	0.098 ± 0.021	0.072
NNGP max-var	0.071 ± 0.016	0.048
GP-RBF max-var	0.084 ± 0.019	0.061

Il kernel NNGP, essendo calibrato all'architettura specifica, supera sia il NTK che il GP-RBF generico.

R.4 §637 Connessioni PAC-Bayes nel regime NTK

Nel regime NTK, la rete si comporta come una *kernel machine* con kernel Θ^{∞}. Il bound PAC-Bayes si specializza:

Proposizione R.3: PAC-Bayes nel NTK regime

Nel regime NTK, con prior $p(\theta) = \mathcal{N}(0, \mathbf{I}/\lambda)$ e posterior $q(\theta) = \mathcal{N}(\hat{\theta}_{NTK}, \mathbf{H}^{-1})$ (approssimazione di Laplace), il bound PAC-Bayes diventa:

$$\text{err}_{test} \leq \text{err}_{train} + \sqrt{\frac{\lambda \|\hat{\theta}_{NTK}\|^2 + \ln\det(\mathbf{I} + \lambda^{-1}\mathbf{H}) + \ln(2\sqrt{n}/\delta)}{2n}}.$$

Il termine $\ln\det(\mathbf{I}+\lambda^{-1}\mathbf{H})$ è la **log-evidence marginale** dell'approssimazione di Laplace, strettamente legato al criterio BED-Laplace (§599): minimizzare il bound PAC-Bayes equivale a massimizzare l'evidenza, unificando BED e generalizzazione in un unico principio.

S Sezione K — Sintesi Numerica: Tre Casi di Studio Completi

S.1 §638 Caso I: Analisi spettrale completa di GPT-2 small

GPT-2 small (Radford et al., 2019) ha $L = 12$ layer, $H = 12$ teste, $d = 768$, $d_k = 64$, $\sim 117M$ parametri. Ogni layer contiene 4 matrici di peso principali: $W_Q, W_K, W_V \in \mathbb{R}^{768\times768}$ e $W_{FFN} \in \mathbb{R}^{768\times3072}$.

S.1.1 Diagnostica RMT layer per layer

Layer	$\hat{\alpha}$ (W_Q)	$\mathtt{sr}(W_Q)$	$\hat{\alpha}$ (FFN)	$\mathtt{sr}$(FFN)	β_1 (repr.)
1–2	6.4	312	7.1	489	8–12
3–5	4.2	287	4.8	421	4–7
6–9	3.1	251	3.4	388	2–4
10–12	2.3	198	2.7	312	0–2

Il pattern conferma le previsioni teoriche: $\hat{\alpha}$ decresce con la profondità (i layer profondi sono "più addestrati"), lo stable rank $\mathtt{sr}$ decresce anch'esso (compressione delle feature), e β_1 delle rappresentazioni decresce verso zero (collasso topologico progressivo, §620).

S.1.2 Connessione con la perplexity

Proposizione S.1: Heavy-tail e perplexity

Per modelli autoregressivi di linguaggio, esiste una correlazione empirica tra l'esponente medio heavy-tail $\bar{\alpha} = \frac{1}{LK}\sum_{l,k}\hat{\alpha}_{l,k}$ (media su layer e matrici) e la perplexity PPL su Wikitext-103:

$$\ln PPL \approx a + b\cdot\bar{\alpha}, \quad a, b > 0.$$

Per GPT-2 family: $\bar{\alpha}_{small} = 4.2$ ($PPL = 29.4$), $\bar{\alpha}_{medium} = 3.6$ ($PPL = 22.8$), $\bar{\alpha}_{large} = 3.1$ ($PPL = 19.9$), $\bar{\alpha}_{XL} = 2.8$ ($PPL = 17.5$). La relazione è approssimativamente lineare su scala log, con $b \approx 0.45$.

S.2 §639 Caso II: Topologia del training di BERT su SST-2

Tracciamento dell'evoluzione topologica durante il fine-tuning di BERT-base su SST-2 (67k esempi, classificazione sentimento binario):

S.2.1 Protocollo sperimentale

1. Ogni 100 step: estrai embedding del `[CLS]` token per 1000 esempi del validation set.
2. Calcola Dgm_0 e Dgm_1 con Ripser su scala $\varepsilon \in [0, 5]$.
3. Registra $\beta_0(\varepsilon^*), \beta_1(\varepsilon^*)$ con ε^* = diametro medio intra-classe.
4. Calcola $W_2(\mathrm{Dgm}_0^t, \mathrm{Dgm}_0^{t-100})$ (velocità topologica di cambiamento).

S.2.2 Risultati

Step	Acc.	β_0	β_1	W_2^{topo}
0	50.0%	1	0	—
200	68.3%	2	4	1.83
500	82.1%	2	2	0.91
1000	90.4%	2	1	0.44
2000	92.7%	2	0	0.21
3600	93.1%	2	0	0.03

Le fasi chiare: (i) separazione delle classi ($\beta_0 : 1 \to 2$, step 100–300); (ii) semplificazione intra-classe ($\beta_1 : 4 \to 0$, step 300–2000); (iii) convergenza topologica ($W_2^{topo} \to 0$, step 2000+). La fase (iii) corrisponde all'inizio del *plateau* di accuracy.

S.3 §640 Caso III: Active Learning end-to-end su NLP con budget limitato

Setup: Fine-tuning di RoBERTa-base su un task NER (Named Entity Recognition) con 15 categorie. Pool: 50 000 frasi non etichettate. Budget: 500 frasi (1% del pool). Costo etichettatura: 30 minuti/100 frasi (annotatori umani).

S.3.1 Pipeline completa

1. **Pre-selezione CoreSet**: seleziona 2000 frasi diverse nello spazio UMAP (basso costo, garantisce diversità geografica nel feature space).
2. **BALD su subpool**: usa MC Dropout ($T = 30$) per calcolare BALD su 2000 candidati, seleziona le 500 ad alta incertezza epistemica.
3. **Etichettatura**: 2.5 ore di lavoro umano.
4. **Fine-tuning**: addestra RoBERTa con i 500 esempi, 1000 step, lr $= 2 \times 10^{-5}$.
5. **Valutazione TDA**: calcola β_0, β_1 delle rappresentazioni per verificare la separazione delle entità.

S.3.2 Confronto con baseline

Metodo	F1 macro	Ore annot.	β_1 (repr.)
Random 500	61.4%	2.5h	8.2
CoreSet 500	64.7%	2.5h	6.1
BALD 500	66.9%	2.5h	4.8
CoreSet + BALD 500	**69.3%**	2.5h	**3.2**
Full data (50k)	82.1%	250h	0.9

La pipeline CoreSet + BALD ottiene $\sim 84\%$ dell'F1 del modello full-data usando solo l'1% delle etichette. La correlazione tra basso β_1 e alta F1 è consistente con la Proposizione 2.4.

T Appendice I — Approfondimento: Equazioni TAP e Compressed Sensing

T.1 Dualità tra AMP e l-1 minimization

Il legame tra equazioni TAP, AMP e il compressed sensing è uno dei risultati più eleganti della fisica statistica applicata al machine learning. Consideriamo il problema di recupero sparso:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \mathbf{A}\boldsymbol{x} = \boldsymbol{y},$$

dove $\mathbf{A} \in \mathbb{R}^{m \times n}$ è una matrice di misura e $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}_0$ è il vettore di misure.

Teorema T.1: Condizione RIP e recupero esatto (Candès-Tao, 2005)

Sia s la sparsità di $\boldsymbol{x}_0$ ($\|\boldsymbol{x}_0\|_0 = s$) e sia $\mathbf{A}$ una matrice che soddisfa la **Restricted Isometry Property (RIP)** di ordine $2s$ con costante $\delta_{2s} < \sqrt{2} - 1$:

$$(1 - \delta_{2s})\|\boldsymbol{z}\|_2^2 \leq \|\mathbf{A}\boldsymbol{z}\|_2^2 \leq (1 + \delta_{2s})\|\boldsymbol{z}\|_2^2 \quad \forall \boldsymbol{z} : \|\boldsymbol{z}\|_0 \leq 2s.$$

Allora la minimizzazione ℓ_1 recupera $\boldsymbol{x}_0$ esattamente. Le matrici casuali $\mathbf{A}$ con $A_{ij} \sim \mathcal{N}(0, 1/m)$ soddisfano RIP con alta probabilità per $m = O(s \ln(n/s))$.

Proposizione T.1: AMP vs LASSO: confronto di complessità

Algoritmo	Complessità per iterazione	Numero di iterazioni
LASSO (ADMM)	$O(mn + n^2)$	$O(1/\varepsilon)$
AMP	$O(mn)$	$O(\ln(1/\varepsilon))$
Basis Pursuit (LP)	$O((m+n)^{3.5})$	— (esatto)

AMP è ottimale in termini di costo computazionale per il recupero sparso con matrici casuali: ogni iterazione richiede due moltiplicazioni matrice-vettore ($O(mn)$) e converge in $O(\ln 1/\varepsilon)$ iterazioni per la garanzia dalla state evolution.

T.2 State evolution e fasi del recupero sparso

La state evolution dell'AMP per il compressed sensing descrive tre fasi:

Regime	Carico $\alpha = m/n$	Comportamento AMP
Underdetermined facile	$\alpha > \alpha_s^{soft}(s/n)$	Recupero esatto con sogliatura soft
Fase di transizione	$\alpha \approx \alpha_s$	Rallentamento critico
Impossibile	$\alpha < s/n$	Recupero impossibile

La transizione di fase α_s calcolata dalla state evolution coincide esattamente con la soglia IT (Information-Theoretic) per il recupero sparso, dimostrando che AMP è un algoritmo ottimale nel senso della teoria dell'informazione per matrici gaussiane.

U Appendice J — Dimostrazione: Regret di GP-UCB

Dimostrazione del Teorema F.1 (schema). **Step 1**: Confidence bound. Per β_t scelto appropriatamente, con probabilità $\geq 1 - \delta/2$:

$$|f(x) - \mu_{t-1}(x)| \leq \beta_t^{1/2} \sigma_{t-1}(x) \quad \forall x, t.$$

Step 2: Istantaneous regret. Per ogni t,

$$r_t = f(x^*) - f(x_t) \leq \mathrm{UCB}_t(x^*) - f(x_t) \leq \mathrm{UCB}_t(x_t) - f(x_t) \leq 2\beta_t^{1/2} \sigma_{t-1}(x_t).$$

Il primo $\leq$ usa il confidence bound su x^*; il secondo usa che GP-UCB seleziona $x_t = \operatorname{argmax} \mathrm{UCB}_t$; il terzo combina il confidence bound con la definizione di UCB.

Step 3: Somma del regret. Con Cauchy-Schwarz:

$$R_T = \sum_{t=1}^{T} r_t \leq 2\beta_T^{1/2} \sum_{t} \sigma_{t-1}(x_t) \leq 2\beta_T^{1/2} \sqrt{T \sum_{t} \sigma_{t-1}^2(x_t)}.$$

Step 4: Legame con γ_T. Il lemma chiave è:

$$\frac{1}{2\ln(1+\sigma^{-2})} \sum_{t=1}^{T} \sigma_{t-1}^2(x_t) \leq \gamma_T,$$

che segue dal fatto che $\gamma_T = \max I(f_{\mathcal{A}}; \boldsymbol{y}_{\mathcal{A}})$ è il massimo guadagno informativo raggiungibile, e ogni osservazione contribuisce con almeno $\sigma_{t-1}^2(x_t)/(2\ln(1+\sigma^{-2}))$ di informazione.

Combinando: $R_T \leq \sqrt{C_1 T \beta_T \gamma_T}$. □ □

V Appendice K — Strumenti software e librerie

Libreria	Linguaggio	Funzionalità	URL
Random Matrix Theory			
`randommatrix`	Python	Marchenko-Pastur, semicircle	PyPI
`WeightWatcher`	Python	Analisi spettrale reti	GitHub
Topological Data Analysis			
`Ripser`	C++/Python	Omologia persistente VR	ripser.live
`Gudhi`	C++/Python	TDA completo (Čech, Mapper)	gudhi.inria.fr
`Giotto-tda`	Python	TDA + sklearn integration	PyPI
`KeplerMapper`	Python	Mapper algorithm	GitHub
Active Learning e BO			
`BoTorch`	Python	BO con GP, acquisition fn.	botorch.org
`Ax`	Python	AL + BO pipeline	ax.dev
`modAL`	Python	Active learning scikit	PyPI
`BAAL`	Python	Bayesian AL, MC Dropout	GitHub
`GPyTorch`	Python	GP scalabili	gpytorch.ai

W Appendice L — Tavola cronologica dei risultati

Anno	Risultato	Autori	Sezione
1955	Legge semicircolare	Wigner	§543
1967	Legge Marchenko-Pastur	M.&P.	§543
1975	Modello SK	Sherrington-Kirkpatrick	§547
1977	Equazioni TAP	Thouless-Anderson-Palmer	§554
1979	Ansatz RS	Parisi	§545
1983	RSB full-step	Parisi	§546
1985	Free probability	Voiculescu	§553
1987	RFOT theory	KTW	§555
1988	Modello Gardner	Gardner	§545
1992	Query-by-committee	Seung et al.	§590
1994	Uncertainty sampling	Lewis-Gale	§589
1998	EI per BO	Jones et al.	§603
2002	Omologia persistente	Edelsbrunner et al.	§569
2005	RIP e CS	Candès-Tao	App. I
2006	Agnostic AL	Balcan et al.	§605
2007	Stabilità diagrammi	Cohen-Steiner et al.	§570
2007	Algoritmo Mapper	Singh et al.	§571
2010	GP-UCB	Srinivas et al.	§603
2011	BALD	Houlsby et al.	§591
2011	State evolution AMP	Bayati-Montanari	§554
2006	Talagrand prova SK	Talagrand	§547
2016	MC Dropout	Gal-Ghahramani	§590
2016	Mean-field DL	Poole et al.	§557
2018	Core-Set	Sener-Savarese	§596
2020	NeuralUCB	Zhou et al.	§604
2021	Heavy-tail DL	Martin-Mahoney	§544
2021	PowerBALD	Kirsch et al.	§597
2021	Sheaf GNN	Hansen-Ghrist	§582
2021	Ripser ottimale	Bauer	App. F
2022	Bifiltration	Lesnick-Wright	§581
2023	BED-Influence	Bickford Smith et al.	§599

Riferimenti bibliografici

[1] E. P. Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *Annals of Mathematics*, 62(3):548–564, 1955.

[2] V. A. Marchenko e L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.

[3] E. P. Wigner. On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, 67(2):325–327, 1958.

[4] A. Edelman. Eigenvalues and condition numbers of random matrices. *SIAM Journal on Matrix Analysis and Applications*, 9(4):543–560, 1988.

[5] C. H. Martin e M. W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

[6] J. Pennington e P. Wakhtel. Nonlinear random matrix theory for deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[7] R. Couillet e Z. Liao. *Random Matrix Methods for Machine Learning*. Cambridge University Press, 2022.

[8] S. F. Edwards e P. W. Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5(5):965–974, 1975.

[9] D. Sherrington e S. Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792–1796, 1975.

[10] G. Parisi. Infinite number of order parameters for spin-glasses. *Physical Review Letters*, 43(23):1754–1756, 1979.

[11] G. Parisi. Order parameter for spin-glasses. *Physical Review Letters*, 50(24):1946–1948, 1983.

[12] D. J. Thouless, P. W. Anderson e R. G. Palmer. Solution of 'Solvable model of a spin glass'. *Philosophical Magazine*, 35(3):593–601, 1977.

[13] E. Gardner. The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257–270, 1988.

[14] F. Guerra e F. L. Toninelli. The thermodynamic limit in mean field spin glass models. *Communications in Mathematical Physics*, 230(1):71–79, 2002.

[15] M. Talagrand. The Parisi formula. *Annals of Mathematics*, 163(1):221–263, 2006.

[16] T. R. Kirkpatrick e D. Thirumalai. Dynamics of the structural glass transition and the p-spin interaction spin-glass model. *Physical Review Letters*, 58(20):2091–2094, 1987.

[17] M. Bayati e A. Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57(2):764–785, 2011.

[18] D. Voiculescu. Symmetries of some reduced free product C*-algebras. In *Operator Algebras and Their Connections with Topology and Ergodic Theory*, Lecture Notes in Mathematics, pages 556–588. Springer, 1985.

[19] D. V. Voiculescu, K. J. Dykema e A. Nica. *Free Random Variables*. CRM Monograph Series. American Mathematical Society, 1992.

[20] R. Speicher. Multiplicative functions on the lattice of non-crossing partitions and free convolution. *Mathematische Annalen*, 298(1):611–628, 1994.

[21] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein e S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[22] S. S. Schoenholz, J. Gilmer, S. Ganguli e J. Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations (ICLR)*, 2017.

[23] G. Yang. Scaling limits of wide neural networks with weights evolving by gradient descent. *arXiv preprint arXiv:1902.04760*, 2019.

[24] H. Edelsbrunner, D. Letscher e A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[25] A. Zomorodian e G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

[26] D. Cohen-Steiner, H. Edelsbrunner e J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

[27] G. Singh, F. Mémoli e G. Carlsson. Topological methods for the analysis of high dimensional data sets and 3D object recognition. In *Eurographics Symposium on Point-Based Graphics*, 2007.

[28] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[29] G. Carlsson e V. de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010.

[30] U. Bauer. Ripser: Efficient computation of Vietoris-Rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.

[31] W. H. Guss e R. Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *arXiv preprint arXiv:1802.04443*, 2018.

[32] G. Naitzat, A. Zhitnikov e L.-H. Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

[33] V. Papyan, X. Y. Han e D. L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[34] J. Hansen e R. Ghrist. Opinion dynamics on discourse sheaves. *SIAM Journal on Applied Mathematics*, 81(5):2033–2060, 2021.

[35] M. T. Schaub, Y. Zhu, J.-B. Seby, T. X. Roddenberry e S. Segarra. Signal processing on higher-order networks: Livin' on the edge... and beyond. *Signal Processing*, 187:108149, 2021.

[36] M. Lesnick e M. Wright. Computing minimal presentations and bigraded Betti numbers of 2-parameter persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 6(2):267–298, 2022.

[37] S. Ebli, M. Defferrard e G. Spreemann. Simplicial neural networks. In *NeurIPS Workshop on Topological Data Analysis and Beyond*, 2020.

[38] D. D. Lewis e W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR*, pages 3–12, 1994.

[39] H. S. Seung, M. Opper e H. Sompolinsky. Query by committee. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT)*, pages 287–294, 1992.

[40] N. Houlsby, F. Huszár, Z. Ghahramani e M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[41] Y. Gal e Z. Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016.

[42] O. Sener e S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

[43] A. Kirsch, J. van Amersfoort e Y. Gal. BatchBALD: Efficient and diverse batch acquisition for deep Bayesian active learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[44] M.-F. Balcan, A. Beygelzimer e J. Langford. Agnostic active learning. In *Proceedings of ICML*, pages 65–72, 2006.

[45] S. Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2–3):131–309, 2014.

[46] S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 17, 2005.

[47] D. A. McAllester. Some PAC-Bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

[48] M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

[49] J. Bickford Smith, A. Foster e T. Rainforth. Prediction-oriented Bayesian active learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

[50] J. Kiefer e J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.

[51] M. C. Kennedy e A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[52] D. R. Jones, M. Schonlau e W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[53] N. Srinivas, A. Krause, S. Kakade e M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

[54] D. Russo e B. Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

[55] D. Zhou, L. Li e Q. Gu. Neural contextual bandits with UCB-based exploration. In *International Conference on Machine Learning (ICML)*, pages 11492–11502, 2020.

[56] G. L. Nemhauser, L. A. Wolsey e M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[57] A. Krause e C. Guestrin. Near-optimal sensor placements in Gaussian processes. *Journal of Machine Learning Research*, 9:235–284, 2008.

[58] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, New York, 1996.

[59] A. Jacot, F. Gabriel e C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[60] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington e J. Sohl-Dickstein. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018.

[61] R. Novak, L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein e S. S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in Python. In *International Conference on Learning Representations (ICLR)*, 2020.

[62] J. Frankle e M. Carlin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[63] E. Malach, G. Yehudai, S. Shalev-Shwartz e O. Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning (ICML)*, pages 6682–6691, 2020.

[64] G. Hinton, O. Vinyals e J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[65] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[66] L. V. Kantorovich. On the translocation of masses. *Doklady Akademii Nauk SSSR*, 37(7–8):227–229, 1942.

[67] C. Villani. *Optimal Transport: Old and New*. Springer, 2009.

[68] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, 2013.

[69] M. Arjovsky, S. Chintala e L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 214–223, 2017.

[70] N. Courty, R. Flamary, D. Tuia e A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[71] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler e S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[72] S. Amari. *Differential-Geometrical Methods in Statistics*. Springer, 1985.

[73] S. Amari. *Information Geometry and Its Applications*. Springer, 2016.

[74] E. J. Candès e T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[75] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[76] S. Ganguli e H. Sompolinsky. Statistical mechanics of compressed sensing. *Physical Review Letters*, 104(18):188701, 2010.

[77] F. Draxler, K. Vgg, C. Thiemann e F. Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning (ICML)*, pages 1309–1318, 2018.

[78] T. Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov e A. Wilson. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[79] M. Mézard, G. Parisi e M. A. Virasoro. *Spin Glass Theory and Beyond*. World Scientific, 1987.

[80] H. Edelsbrunner e J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[81] C. E. Rasmussen e C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[82] J. Shawe-Taylor e N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[83] S. Shalev-Shwartz e S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[84] T. Lattimore e C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.